COMPUTER SCIENCE, TECHNOLOGY AND APPLICATIONS

Neural Network Driven Artificial Intelligence

Decision Making Based On Fuzzy Logic

Bahman Zohuri
Masoud Moghaddam

NOVA

VISIT…

LANZAROTE Caliente.com

COMPUTER SCIENCE, TECHNOLOGY AND APPLICATIONS

NEURAL NETWORK DRIVEN ARTIFICIAL INTELLIGENCE

DECISION MAKING BASED ON FUZZY LOGIC

Computer Science, Technology and Applications

Additional books in this series can be found on Nova's website
under the Series tab.

Additional e-books in this series can be found on Nova's website
under the eBooks tab.

Mathematics Research Developments

Additional books in this series can be found on Nova's website
under the Series tab.

Additional e-books in this series can be found on Nova's website
under the eBooks tab.

COMPUTER SCIENCE, TECHNOLOGY AND APPLICATIONS

NEURAL NETWORK DRIVEN ARTIFICIAL INTELLIGENCE

DECISION MAKING BASED ON FUZZY LOGIC

BAHMAN ZOHURI
AND
MASOUD MOGHADDAM

nova science publishers
New York

Library of Congress Cataloging-in-Publication Data

ISBN: 978-1-53612-138-4 (eBook)

Published by Nova Science Publishers, Inc. † New York

This book is dedicated to my daughter Natalie Zohuri (MBA)

Bahman Zohuri

This book to Hanna, the love of my life who has been a great support for me with her true love

Masoud Moghaddam

CONTENTS

PREFACE

With today's growing information and the overloading of its volume, it is becoming tremendously difficult to analyze the huge amounts of data that contain the information and which makes it very strenuous and inconvenient to introduce an appropriate methodology of decision making fast enough to the point that it can be considered as real time. The demand for real time processing information and related data – both structured and unstructured – is on the rise and consequently makes it harder and harder to implement correct decision making at the enterprise level to keep the organization robust and resilient against either manmade threats or natural disasters.

Today's campaign against any cyber attack has put a huge demand on cyber security and on information security personnel at different levels of any organization. Therefore, processing incoming data as sets of information becomes more and more critical. Furthermore, the data are often, imprecise and will include both quantitative and qualitative elements. For these reasons, it is important to extend traditional decision making processes by adding intuitive reasoning, human subjectivity and imprecision. To enhance this process of decision making, these authors have taken an unorthodox approach by applying a new growing technology known as a neural network as part of driving infrastructure for an artificial intelligence system to take over from human operation in order to satisfy the demand for real time decision making.

To deal with uncertainty, vagueness, and imprecision, Lofti A. Zadeh introduced fuzzy sets and fuzzy logic. In the present book, fuzzy classification is applied to extend portfolio analysis, scoring methods, customer segmentation and performance measurement, and thus to improve managerial decisions. As an integral part of the book, case studies show how fuzzy classification with its query facilities can extend customer equity, enable mass customization, and refine marketing campaigns.

This book follows up on our first book under the title of *Business Resilience System (BRS): Driven through Boolean, Fuzzy Logics and Cloud Computation: Real and Near Real Time Analysis and Decision Making System 1st ed. 2017 Edition.* In this book, we have decided to expand on BRS by talking about the Artificial Neural Network (ANN) and building a foundation for the readers who are new to the field of ANN and how such an approach can lead us to smarter and more autonomous BRS as well as how the architecture of ANN works.

Although artificial neural networks have simpler processing units than typical Central Processing Units (CPUs), their unique form of parallel processing and tremendous number of

interconnections make them incredibly versatile problem solvers. This far-reaching source gets you in on the ground floor of this state-of-the-art technology, giving you a complete overview of neural network capabilities that drives artificial intelligence, limitations, components, and applications in diverse fields.

An important aspect of this approach by us is to examine biological neural systems and how artificial neural networks are based on them and driven by them as well. Key areas discussed include structural diversity, temporal aspects, origins of artificial neural systems, brain structure and function, biological nerve cells, synapses, fixed and random components in the brain's neural networks, and how biological systems really compare to computational neural networks.

For us to deliver such a momentum of knowledge in neural networking to our readers, we have reached to experts in the field and with their permission have quoted their notes, lectures and their presentations from their websites at various universities and industries. Our many thanks go to experts like Professor Ingrid Russell of the University of Hartford for her permission to use from the Collegiate Microcomputer (See some ideas in Chapter 3), Kiyoshi Kawaguchi for his lectures, and Professor David Beeman of the University of Colorado, Department of Electrical and Computer Engineering (See Chapter 6) on Computational Neuroscience. In addition, some information that is used in Chapter 5 was permitted by Dr. David Leech Anderson, Professor of Philosophy at Illinois State University as well as art work provided by “graphic artists” at The Mind Project.”

Every chapter yields new insights into the potential of artificial neural networks (ANNs) for creative hardware implementations and the software running them. Plus, some examples of applications that can be run on off the self neural network software packages make it easy to put these networks right to work in your desktop computer.

This book is written as a means of introduction for both artificial and biological neural networks to disparate readers and audiences, comprising engineers and biologists as well as CEOs in charge of retail establishments, banking and manufacturing with huge production lines and business professionals as well as others. It acts as more than a simple introduction; this book makes an excellent compendium and builds an essential foundation for those who have strong interests to peruse the field of neural networking drive/ artificial intelligence and robotic systems of the future from the autonomy perspective.

This can be done by focusing on the neural network paradigm, which is of equal use to beginners and professionals utilizing artificial neural networking as well as the biological neural networking implemented into robots of the near future, and perhaps to make the matrix imagination a reality.

Artificial Neural Networks (ANNs) are the wave of the future, beginning anywhere from being capable of doing everything from translating financial data into financial predictions, to mapping visual images to robotic commands, to classifying medical images for diagnostic tests and so on. This is why a book like this is essential reading for computer programmers, hardware and software engineers and specialists, managers, biologists, and other professionals who want to stay in the vanguard of their fields.

B. Zohuri
M. Moghaddam
Albuquerque, New Mexico, 87111

ACKNOWLEDGMENT

I am indebted to the many people who aided me, encouraged me, and supported me beyond my expectations. Some are not around to see the results of their encouragement in the production of this book, yet I hope they know of my deepest appreciations. I especially want to thank my friend Bill Kemp, to whom I am deeply indebted, has continuously given his support without hesitation. He has always kept me going in the right direction.

Above all, I offer very special thanks to my late mother and father, and to my children, in particular, my son Sasha. They have provided constant interest and encouragement, without which this book would not have been, written. Their patience with my many absences from home and long hours in front of the computer to prepare the manuscript are especially, appreciated.

Our special thanks go to Professor David Leech Anderson of Illinois State University and Professor David Beeman of University of Colorado.

About this Document

This section describes the document's purpose, scope, and audience, lists documents that provide additional, related information, and provides definitions of terms.

Purpose

The purpose of this document is to describe Dimensional Analysis, Similarity and Modeling Methods.

About the Authors

Dr. Bahman Zohuri is currently at Galaxy Advanced Engineering, Inc., a consulting company that he started himself in 1991 when he left both the semiconductor and defense industries after many years working as a chief scientist. After graduating from the University of Illinois in the field of Physics and Applied Mathematics, he joined Westinghouse Electric Corporation, where he performed thermal hydraulic analysis and natural circulation for the Inherent Shutdown Heat Removal System (ISHRS) in the core of a Liquid Metal Fast Breeder Reactor (LMFBR) as a secondary fully inherent shut system for secondary loop heat exchange. All these designs were, used for Nuclear Safety and Reliability Engineering for a Self-Actuated Shutdown System. He designed the Mercury Heat Pipe and Electromagnetic Pumps for Large Pool Concepts of LMFBR for a heat rejection purpose for this reactor around 1978 where he received a patent for it. He then was, transferred to defense division of Westinghouse later, where he was responsible for the dynamic analysis and method of launch and handling of an MX missile out of a canister. He later on was a consultant at Sandia National Laboratory after leaving the United States Navy. Dr. Zohuri earned his Bachelor's and Master’s degrees in Physics from the University of Illinois and his second Master’s degree in Mechanical Engineering as well as his Doctorate in Nuclear Engineering from the University of New Mexico. He has been, awarded three patents, and has published 26 textbooks and numerous other journal publications.

Recently, he has been involved with cloud computation, data warehousing, and data mining using fuzzy and Boolean logic

Mr. Masoud Moghaddam obtained his Master’s degree in Business Administration and has been a programmer and developer for over 25 years now. He became an ISP (Internet Service Provider) in the early days of internet popularity, has great knowledge and experience in IP-driven and web applications, intranet services, cyber security, database programming and Graphical User Interface (GUI) design.

His years of experience and knowledge of networking and cyber security involved him in many projects in the past 20 years. In addition, he has managed many total solution projects in networking and software development while he was working as the director of IT for Galaxy Advanced Engineering; he was also involved in many scientific and technical projects to date.

Masoud also has deep views about the future of networking, Artificial Intelligence (AI) and organic computing, where he is currently working on the fundamentals and methodology of those new concepts.

The two authors as part of their collaboration recently have published a new book through the Springer Publishing Company under title of:

Business Resilience System (BRS)
Driven Through
Boolean Fuzzy Logics and Cloud Computation
Real and Near Real Time Analysis and Decision Making System

Chapter 1

KNOWLEDGE IS POWER

The phrase “**scientia potentia est**” (or “**scientia est potentia**” or also “scientia potestas est”) is a Latin aphorism meaning “knowledge is power.” It is commonly, attributed to Sir Francis Bacon, although there is no known occurrence of this precise phrase in Bacon's English or Latin writings. However, the expression “ipsa scientia potestas est” ('knowledge itself is power') occurs in Bacon's Meditationes Sacrae (1597). The exact phrase “scientia potentia est” was written for the first time in the 1668 version of the work Leviathan by Thomas Hobbes, who was secretary to Bacon as a young man. The related phrase “sapientia est potentia” is often translated as “wisdom is power.” Though its meaning varies from author to author, the phrase often implies that with knowledge or education, one's potential or abilities in life will certainly increase. Having and sharing knowledge is widely recognized as the basis for improving one's reputation and influence, thus power. This phrase may also be used as a justification for a reluctance to share information when a person believes that withholding knowledge can deliver to that person some form of advantage. Another interpretation is that the only true power is knowledge, as everything (including any achievement) is derived from it. “Knowledge is Power” is a popular phrase. Knowledge provides us with the power to help others, in a variety of ways. It is also something that is good for our own self esteem. In addition, knowledge imbues us with authority and enables us to act and interact with others in a more moral way. Power refers to the ability or capability to do something. It can also include the strength to influence the actions of others. This can mean physical strength or persuasive power.

1.1. INTRODUCTION

‘Knowledge is power’ is a popular proverb. It means that knowledge is more powerful than physical strength and no great work can be done without knowledge. Knowledge is a powerful factor that empowers people achieves great results. The more knowledge a person gains, the more powerful he becomes.

This proverb means that ‘true power comes from knowledge’. No individual or nation can prosper in life without knowledge. In sum, the proverb means that knowing things gives us power. There is no end to knowledge. There is no limit to what a person can learn. Even big problems can be solved if we have the knowledge of solving it.

By knowledge of science man has conquered nature. Development is possible by knowledge and not by physical strength. For example:

- A teacher without through knowledge cannot teach his student well.
- A student without any knowledge of his subject cannot pass the examination.
- A doctor without knowledge in surgery cannot be a successful doctor.
- A pleader without proper legal knowledge cannot argue well in favor of his client.
- An Artificial Intelligent (AI) without proper information and data cannot be a true AI

The last example statement is focus of this book and subject of our continuous discussion of reliable Business Resilience System (BRS) [1].

Knowledge of information that comes from various trusted data is the main core of a smart BRS and fundamental requirement of infrastructure for an Artificial Intelligence (AI) is support of such a Business Resilience System.

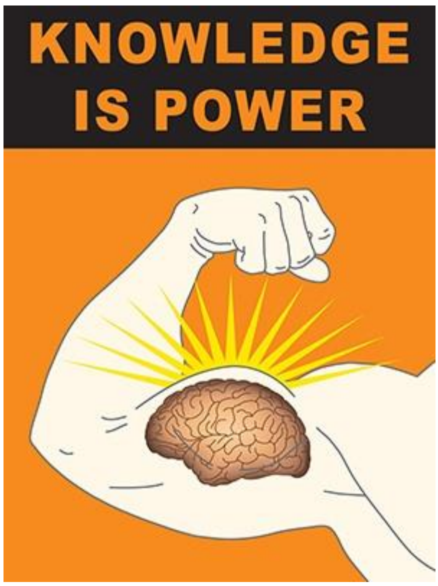


Figure 1-1. Imaginary Presentation of Knowledge is Power.

The expansion of this proverb is summarized as follows:

- Knowledge is considered superior to physical strength in gaining success. A society or community that is devoid of knowledge is considered backward, even if they are a physically strong group. Many physically powerful nations were defeated by nations having greater intelligence and knowledge.
- Knowledge helps human beings to utilize the various forces of nature for the benefit of humanity. The rise of human beings as the most powerful living-beings on planet is only due to the knowledge and the proper application of knowledge.
- Knowledge plays a vital role in every sphere of human life and activity. Knowledge has helped in the advancement and development of civilization and culture. The

application of knowledge has led man to the path of progress. We can all benefit from learning new things.

And importance of the proverb is:

1. **Study:** the proverb motivates us to study and acquire knowledge.
2. **Education:** education is lifelong.
3. True power: real power comes from being educated and knowledgeable, not from violence or controlling others.
4. **The joys of learning:** this proverb encapsulates the joy inherent in learning new things.

A person who takes decisions based on understanding, knowledge, and facts is bound to have edge over others. Well informed and well thought decisions carry calculable risks and open the doors for massive success. Humanity's insatiable hunger for knowledge makes it active and dynamic. This hunger cannot be satisfied until the unknown is unveiled.

Twin is the purpose of earning knowledge. Knowledge gives us joy and power. Knowledge empowers us to face the challenges of life. It emboldens us to bear with the stark realities of life with a calm of mind. Knowledge also gives us immense power by dint of which we can master all. It makes us immensely inventive.

The following examples explain how knowledge empowers people:

- The invention of almost every modern device has involved some form of technology. Technology is the practical application of scientific knowledge. Scientific knowledge is the key to major discoveries.
- Mathematical knowledge empowers a person to calculate and keep proper record to their valuables, resources and other things. A business person is able to keep record of his transactions. The knowledge of mathematics has empowered the business people to forecast the sales growth in number format.
- Geographical knowledge enables a person to locate any place in this world. People can plan and tract their movement easily.
- History makes us aware of past events. We learn a lot when we study the life of other people.
- Knowledge of economics enables us to contribute towards both individual economic prosperity and national economic growth.

Knowledge can include skills, experience and education. At best, it includes all three! Knowledge can be practical knowledge of the kind used by an engineer or a carpenter, or it can be more abstract knowledge of the kind that a mathematician uses. Knowledge helps us to find the way to solve a variety of issues.

However, power of knowledge often misleads us. The world has witnessed several wars and battles. This is undesirable. Let us hope that knowledge will make us powerful to promote our peace and prosperity in life.

1.2. Reasons Why Knowledge Is Power

There are five reasons, why knowledge is power, and they are listed here as:

1. **Knowledge liberates us.**
   Knowledge sets us free, and makes us less dependent on others. Freedom is essential for real power. Of course, being truly free means that we do not use our power to control other people against their will.
2. **Respect.**
   True knowledge commands more respect that mere empty authority within a hierarchy ever could. If we have knowledge, we can direct others' decisions and help them to enhance their lives. Having knowledge about a relevant subject imbues us with authority. No matter who we are, or how old we are, if we have knowledge that is useful to other people, then those people will respect us.
3. **Self esteem.**
   Possessing knowledge can really give us a feeling of self fulfillment and confidence. Knowledge is something that – no matter how many trials we come across in our life – we can always fall back on. In addition, if we find ourselves facing a trial in life, knowledge can enable us to find a solution to the issue that boosts our self esteem even further. What could be more of a confidence boost than knowing that we used our own skills and knowledge to surmount one of life's challenges?
4. **Positivity.**
   The process of seeking and finding knowledge teaches us to have a positive attitude about life. It teaches us to be motivated, determined, engaged with the world and self reliant. It also fills us with enthusiasm and joy – after all, humans love learning new things and the process of finding out new facts is a wonderful end in itself.
5. **Morality.**
   When we have knowledge, we can act more morally. Possessed of all the facts and the relevant skills, we can put our desire to help others into practice much better than we could do if we had less knowledge. For example, if we have some money that we wish to donate to charity, knowing facts about how that money could best be used will enable us to help the greatest number of people with it.

There is no denying that there are several convincing arguments for the notion that 'knowledge is power', and it is always best to use our power for good.

1.3. Knowledge Is Power? Those Days Are Long Gone

Knowledge was power, back when knowledge was not easily available or disseminated. If you lived in the 1600s and wanted to be a stonemason, you'd start off as a master's apprentice. Instead of paying you, he'd teach you his trade. He could do this because he had the knowledge you could not get elsewhere. He had power. You? Not so much.

Then movable type came long. Printing presses started cranking out books, a few at first, and they were rare and expensive. Over time, with the printed word, knowledge was no longer so easily contained or controlled, and it was transformative. Published information eventually became available on every imaginable topic. Publications grew less expensive, so that average people could actually afford to buy books. Schools began to proliferate. More children learned to read, and new opportunities appeared. You did not have to be a stonemason if you did not want to. You could be an airplane pilot instead.

Knowledge was spreading, and those who possessed it were no longer in the minority—which meant they were not as powerful as they used to be. Eventually came radio, TV, and the Internet. The more information we had available, the more people developed highly specialized knowledge—something they believed would give them power.

Today, when most people in developed nations have easy access to information from around the world, communication costs are cheap, and we have plenty of education options, even specialized knowledge does not necessarily give you power. In fact, you do not even have to go to great lengths to develop a specialized knowledge anymore. Just do a Google search and borrow someone else's.

In an era of widespread, inexpensive communications, knowledge spreads way too fast for it to hold power for long. So there's no point in trying to cram tons of it into your head on the assumption it will make you special and give you power. It will not.

1.4. There Is More to the Knowledge Is Power

Us as authors, think that there more dimensional to simple proverb of "knowledge is power." In order to have knowledge, you need information and in order to have information you need trusted data.

Now imagine that if you could have these data transferred to you in real-time and your information gets updated accordingly as fast as your Internet speed, then you knowledge can be ahead of your competitor and that is the power you need to conduct your day-to-day operation and decision making within your organization or enterprise. This the part of knowledge management, which is important to the success of your company.

Managers are bombarded with an almost constant stream of data every day. According to David Derbyshire, "Scientists have worked out exactly how much data is sent to a typical person in the course of a year - the equivalent of every person in the world reading 174 newspapers every single day" (Derbyshire, 2011, p. 1) [2].

This overload of data is making knowledge management increasingly more important. Three key reasons why actively managing knowledge is important to a company's success are [3]:

1. Facilitates decision making capabilities,
2. Builds learning organizations by making learning routine, and,
3. Stimulates cultural change and innovation.

1.4.1. Facilitates Decision -- Making Capabilities

Data can offer managers a wealth of information but processing overwhelming amounts can get in the way of achieving high-quality decisions. GE's Corporate Executive Council (CEC) is an example of how one company put a knowledge management system in place to help executives cut through the noise, share information, and improve their decision making. The CEC is composed of the heads of GE's fourteen major businesses and the two-day sessions are forums for sharing best practices, accelerating progress, and discussing successes, failures, and experiences (Garvin, 2000, p. 195) [4]. While information overload or needing knowledge from people in other parts of the company for decision making can handicap managers, putting in place knowledge management systems can facilitate better, more informed decisions.

However what Garvin says in his book, is real fact and that is "Today's business leaders must always know what their stakeholders are thinking--be they customers, employees, constituents, or competitors--and act upon that information in a timely and appropriate manner. How companies collect, decipher, and utilize this knowledge, in fact, may be the real determinant of their long-term viability. Harvard Business School professor David A. Garvin's Learning in Action authoritatively dissects these activities as practiced by so-called learning organizations, then clearly outlines the steps necessary to build one of them. "Sweeping metaphors and grand themes are far less helpful than the knowledge of how individuals and organizations learn on a daily basis," Garvin writes. "The key to success is mastery of the details, coupled with a command of the levers that shape behavior." His book's core offers a practical examination of the three primary routes to corporate learning: collecting intelligence from outside sources (via interview and observation, for example); accumulating data through targeted actions (such as post-project reviews and special programs); and experimenting with alternative outcomes by manipulating variables (including prototype creation and exploratory design testing). Combining research from myriad fields, detailed studies of successful models such as Xerox and the U.S. Army, and snapshots of specific practices at additional firms such as Intel and Wal-Mart, he succeeds in providing "a broad, integrated view of the topic that is grounded in scholarship."

In his book, Learning in Action: A Guide to Putting the Learning Organization to Work, author David Garvin (2000) notes, "To move ahead, one must often first look behind" (p. 106). The U.S. Army's After Action Reviews (AARs) are an example of a knowledge management system that has helped build the Army into a learning organization by making learning routine. This has created a culture where everyone continuously assesses themselves, their units, and their organization, looking for ways to improve. After every important activity or event, Army teams review assignments, identify successes and failures, and seek ways to perform better the next time (Garvin, 2000, p. 106) [4]. This approach to capturing learning from experience builds knowledge that can then be used to streamline operations and improve processes.

1.4.2. Stimulates Cultural Change and Innovation

Actively managing organizational knowledge can also stimulate cultural change and innovation by encouraging the free flow of ideas. For example, GE's Change Acceleration

Process (CAP) program includes management development, business-unit leadership, and focused workshops. CAP was created to not only "convey the latest knowledge to up-and-coming managers" but also "open up dialogue, instill corporate values, and stimulate cultural change" (Garvin, 2000, p. 125). In this complex, global business environment, these types of knowledge management programs can help managers embrace change and encourage ideas and insight, which often lead to innovation, even for local mom and pop business owners.

Considering, how our today technologies is influencing and changing our present knowledge management direction to tomorrow's robust and more resilience form, we look at the banking organization and their Customer Relationship Management (CRM). Today's more customers are using mobile. They've started using their mobile devices almost like their wallets, and we know that these devices are going to be used increasingly more often over time. So it is very applicable for customers, especially for depositing checks. There's a technology out there called Remote Deposit Capture (RDC) that allows a customer to take a picture of a check. The check is processed and automatically deposited into the customer's account.

So the point is the knowledge management is not about "pushing" information to the desktop. It is about a 360-degree view of the entire customer journey or episode. It is about what is happening to the entire customer in all channels they're using. We need to aggregate all of that information, including the temperament of the customer to agent, with a recommended next best action. At least this what Mr. David Bradshaw, Vice President, Banking Operations, ING DIRECT Canada.

Therefore, knowledge is crucial to understanding the customer journey. When a customer calls about an issue they're trying to resolve, they may already have gone through multiple channels, and they feel that you should understand their level of frustration about that already. So it's important for us to aggregate all of that data across the organization so that we can do whatever is required to address customer needs.

1.4.3. Bottom Line

Fortune 500 companies lose roughly "$31.5 billion a year by failing to share knowledge" (Babcock, 2004, p. 46) [5], a very scary figure in this global economy filled with turbulence and change. Actively managing knowledge can help companies increase their chances of success by facilitating decision making, building learning environments by making learning routine, and stimulating cultural change and innovation. By proactively implementing knowledge management systems, companies can re-write the old saying, "Change is inevitable, growth is optional" to "Change is inevitable, growth is ***intentional***."

Many knowledge management experts point to the events of Sept. 11, 2001, as the ultimate reason that information-sharing systems are necessary. The FBI has drawn criticism for ignoring intelligence that could have thwarted the terrorist attacks and for an apparent lack of communication with the CIA. According to congressional hearings, government agencies ignored many other signs—some going back as far as 1994.

It is a dramatic example of the failure of knowledge management. In an agency such as the FBI, where the sharing of information can save lives, such lapses can mean tragedy.

In the business world, they can bring huge financial losses. Fortune 500 companies lose at least $31.5 billion a year by failing to share knowledge, according to International Data

Corp. (IDC), a Framingham, Mass-based market intelligence and advisory firm in the IT and telecommunications industries.

It's not that companies and organizations are not trying. Since knowledge management became all the rage in the high-flying 1990s, companies have poured tremendous resources into knowledge management technology that has failed miserably or shown little results. Businesses sank $2.7 billion into new systems in 2002, according to the IDC, which estimates that number to rise to $4.8 billion in 2007. The federal government will boost knowledge management spending from $820 million in 2003 to $1.3 billion by 2008, largely for homeland security requirements, according to INPUT, a Reston, Va., market research company.

Will these expensive new attempts work? Only if they take clues from past failures and develop a different approach, experts say. The reasons for failure are many and varied, but two factors seem common: Technology is too complicated, and, perhaps most important, organizations do not give enough consideration to the barriers human nature poses to information sharing.

Steve Denning of IBM corporation, claims that, there are "Ten Things You Need To Know About Managing Knowledge," and we quote him directly here.

> A little learning is a dangerous thing;
> Drink deep, or taste not the Pierian spring
> Alexander Pope

He states that: [5]

How does an organization decide what to spend on knowledge? What is the value of investments in R&D or knowledge management? What can an organization do to improve the effectiveness of these investments? Finding answers to these questions isn't easy because the amount of spending on knowledge typically doesn't correlate with results.

Issues in the Private Sector

In the private sector, conventional measures of R&D effectiveness—for instance, the amount of spending or number of patents—don't answer those questions or reliably predict market value. Booz & Company's annual innovation reports repeatedly state: "Spending more on R&D won't drive results. The most crucial factors are strategic alignment and a culture that supports innovation." Yet it's hard to measure culture or alignment, let alone link them to profitability or market value. Simple formulaic approaches can give some weird results.

In a profit-oriented context, investments in knowledge like R&D or knowledge management are easy targets when firms face quarterly earnings pressure. Cuts can yield immediate increases in profit, but the impact of those cuts on long-term sustainability can be devastating, as Dell [DELL], Hewlett Packard [HP] and Sony [SNE] have discovered.

At the other end of the spectrum, some firms, including some of the big pharmaceutical companies, spend vast amounts of resources on R&D only to find that returns on investment are meager even in the long run. Still other firms, like Apple [AAPL], which spend little on basic research, are able to generate repeated and hugely profitable innovations.

Issues in the Public Sector

Similar issues apply in the public sector. Mahmoud Mohieldin, Managing Director for the World Bank, has written recently:

The World Bank's goal is to reduce poverty, with services that combine financing and technical expertise. The Bank invests more than $600 million annually in knowledge services, including original data collection, research, and technical assistance on topics ranging from education to health, infrastructure, communications, government reforms. In addition, the Bank seeks to promote learning through its project financing, testing new approaches to deliver public services and seeking to understand what works best to alleviate poverty. Total spending on knowledge services -- through the Bank's loans, budget and partnership activities -- is approximately $4 billion per year. It's banking with ideas.

Is $4 billion the right amount for the World Bank to be spending on knowledge services? Should it be spending more—or less? Could it be getting more value from its investment? Does it have the systems and resources in place to get the best value on its investments in knowledge services?

Ten Principles for Managing Knowledge

To address these questions, it is useful to begin with the basic principles for understanding the management of knowledge.

1. **The amount of money that could be spent on accumulating knowledge is infinite:** Knowledge is in principle limitless. The US Library of Congress has over 100 million items: it could be argued that all of them contain some knowledge of some kind that might be useful to someone at some time now or in the future. The knowledge is there "just in case." Any organization could spend its entire resources on knowledge and still not have exhausted the possibilities of accumulating knowledge.
2. **Knowledge has no value per se:** Knowledge acquires value from use. Vast amounts of money can be spent on storing knowledge for potential use in the future without ever leading to the creation of any actual value.
   One large consulting firm offered incentives to its staff to input knowledge objects into a computerized knowledge base. After a few years, it had assembled around 1.6 million knowledge objects. However the knowledge base was of little value as it wasn't being used.
3. **Spending on knowledge has negative value if organization does not use it:** Knowledge is only useful to those willing and able to learn, as Professor Martine Haas of the Wharton School showed in her research on the impact of access to knowledge on the performance of teams in the World Bank.
   Teams operating in a context that encouraged learning and innovation improved their performance as compared to teams that didn't have access to knowledge. But when teams were operating in environments where the result of their work was predetermined, with little flexibility allowed to the team to adapt and innovate, performance didn't improve along with improved access to knowledge. In this constrained setting, access to knowledge just slowed the team down. Without a desire to learn, improved access to knowledge was a liability, not an asset.

4. **Institutional knowledge may serve as blinders to effective action:** Practices within an organization which are viewed as institutional knowledge may prevent an organization from getting access to, and using, the knowledge it really needs. In one study, Professors Martine Haas and Morten Hansen examined the use of internal knowledge systems by teams of consultants in one of the big four consulting firms trying to win sales bids.
   In order to encourage the staff to make use of the large-scale knowledge resources that had been assembled, staff were offered incentives to incorporate resources from the knowledge base into their work, particularly bids for new engagements. A study then measured to what extent these teams accessed electronic documents and how much they sought personal advice from other consultants in the firm. The expectation was that accessing more knowledge would be helpful. But the study showed that the more internal electronic databases were consulted by these teams the more likely they were to lose the bid. Use of the knowledge base resulted in cookie cutter proposals that were less responsive to the client's needs than proposals that were closely oriented to the clients' specific needs.
5. **The most valuable knowledge increasingly lies outside the organization:** Fifty years ago, all knowledge was inside the organization. Today the most valuable knowledge may be outside the organization. The fact that Apple was able to produce a mobile phone in 18 months from a standing start demonstrates how far knowledge in some fields has become a commodity. The locus of value creation has shifted closer to the ultimate customer and the resulting knowledge can diffuse rapidly. As John Hagel, John Seely Brown and Lang Davison point out in their book, The Power of Pull (2010), mastering knowledge flows is at least as important as funding in-house R&D. In this world, success comes from:
   - Accessing resources and people with know-how, whether those resources and people are outside the firm or within.
   - Attracting people and resources to come to you and collaborate in generating more value.
   - Accomplishing results based on these knowledge flows, by facilitating partnerships based on collaboration and reliable production.
6. **Knowledge can require deep expertise to access it:** Just as knowledge can be acquired, so it can be lost. Thus organizations can lose key expertise so that it no longer has the expertise needed to effectively access knowledge it once had. Consider the experience of Dell in manufacturing computers from The Innovator's Prescription.
   ASUSTeK started out making the simple circuit boards within a Dell computer. Then ASUSTeK came to Dell with an interesting value proposition: "We've been doing a good job making these little boards. Why don't you let us make the motherboard for you? Circuit manufacturing isn't your core competence anyway and we could do it for 20% less."
   Dell accepted the proposal because from a perspective of making money, it made sense: Dell's revenues were unaffected and its profits improved significantly. It was able to let go staff who would have been needed on design and

development. On successive occasions, ASUSTeK came back and took over the motherboard, the assembly of the computer, the management of the supply chain and the design of the computer. In each case Dell accepted the proposal because from a perspective of making money, it made sense: Dell's revenues were unaffected and its profits improved significantly. However, the next time ASUSTeK came back, it wasn't to talk to Dell. It was to talk to Best Buy and other retailers to tell them that they could offer them their own brand or any brand PC for 20 percent lower cost. As a result, Dell no longer had the expertise needed to compete as a computer manufacturer: it had become little more than a brand. Dell no longer had the expertise needed to access the knowledge it once had.

7. **The deep expertise needed to access knowledge can be lost:** Decades of outsourcing manufacturing have left much of U.S. industry without the means to invent the next generation of high-tech products that are key to rebuilding its economy, as noted by Gary Pisano and Willy Shih in their 2009 HBR article, "Restoring American Competitiveness" They write:
   "The decline of manufacturing in a region sets off a chain reaction. Once manufacturing is outsourced, process-engineering expertise can't be maintained, since it depends on daily interactions with manufacturing. Without process-engineering capabilities, companies find it increasingly difficult to conduct advanced research on next-generation process technologies. Without the ability to develop such new processes, they find they can no longer develop new products. In the long term, then, an economy that lacks an infrastructure for advanced process engineering and manufacturing will lose its ability to innovate."

Pisano and Shih list the industries of industries that are "already lost" to the USA:

> "Fabless chips"; compact fluorescent lighting; LCDs for monitors, TVs and handheld devices like mobile phones; electrophoretic displays; lithium ion, lithium polymer and NiMH batteries; advanced rechargeable batteries for hybrid vehicles; crystalline and polycrystalline silicon solar cells, inverters and power semiconductors for solar panels; desktop, notebook and netbook PCs; low-end servers; hard-disk drives; consumer networking gear such as routers, access points, and home set-top boxes; advanced composite used in sporting goods and other consumer gear; advanced ceramics and integrated circuit packaging.

Their list of industries "at risk" is even longer:

> Just as countries can lose knowledge, so organizations can be at risk of losing knowledge by eliminating senior staff with deep expertise and instead relying on consultants or staff hired on fixed-term assignments: the organization may no longer have either the expertise in-house or the deep expertise needed to access knowledge outside the organization to achieve its goals.

8. **The value of knowledge lies in improved outcomes for external customers or stakeholders.** Given that knowledge has no value in itself and has value only when it is put to use, knowledge should not be evaluated merely as an output. Instead, knowledge should be viewed as valuable only when it results in an improved outcome for some ultimate customer or stakeholder. In knowledge activities, as elsewhere in the 21st Century economy, outcomes are key, not outputs.
9. **What constitutes an improved outcome depends on the organization's strategy:** Outcomes must be assessed in terms of the organization's strategy. A knowledge investment that is sensible in an organization with one kind of strategy might make no sense in another with a different strategy.

   In the private sector, the initial questions are: who are the firm's core customers? Is the knowledge generating more value for those customers or delivering it sooner? Within what time frame will the improved outcomes occur?

   In the public sector, the initial questions are: who are the core stakeholders? Is the knowledge generating more value for those stakeholders or delivering it sooner? Within what time frame will the improved outcomes occur?

   Many public sector organizations have failed to reach closure on what their strategy really is, who their core stakeholders are or what would constitute increased value for them: such organizations often try to cater to the supposed needs of a multitude of stakeholders, and end up satisfying few or none of them in any systematic fashion. Evaluating knowledge activities in such organizations will be particularly problematic, unless and until they can make further progress on clarifying their strategy.

   The World Bank for instance has an official mission "to fight poverty with passion and professionalism for lasting results; to help people help themselves and their environment by providing resources, sharing knowledge, building capacity and forging partnerships in the public and private sectors."

The mission statement leaves open several different interpretations:

10. **The Bank as a lending organization for national economic development:** For much of its life the World Bank has been a bank that lends money for development as its main role, helping "developing countries" become "developed countries," with poverty alleviation being something happens as part of that process. Within this framework, once a country is "developed," the World Bank's job is done: it can turn its attention to the remaining "developing countries." Some observers believe that the World Bank's management systems and operations continue to function as though this is really still its mission, despite the tension with the current official mission statement. If this view is correct, the World Bank would over time have progressively fewer and fewer countries to assist and would eventually go out of business when all countries are "developed."

    The World Bank as a poverty reduction organization: An alternative interpretation is to take the official mission statement literally. On this basis, the World Bank's mission is to relieve poverty, either by helping poor countries, or

by helping poor people in all developing countries or even by helping poor people wherever they are.

The World Bank as a provider and financier of public goods: Yet another view is that the World Bank is an organization uniquely equipped to lead global action and provide knowledge on selected global issues, such as climate change, poverty reduction and economic statistics.

11. **Outcomes need to be measured against the organizational strategy:** Nothing can be managed unless and until it is measured. Because knowledge per se has no value in itself, it has to be measured in relation to the strategy it is intended to accomplish.

    In the case of a private sector organization, resources spent on the creation or sharing of knowledge should be evaluated in relation to the outcomes expected from the products and services that the firm is generating, either now or in the future.

    In the case of a public sector organization, resources spent on the creation or sharing of knowledge need to be evaluated in relation to the outcomes for the core stakeholders expected to benefit from the strategy.

Particular issues in evaluating knowledge services:

12. **Mission clarity:** For some organizations, the most pressing issue may be the clarification of strategy. Having an unclear strategy has at least two implications for evaluating the management of knowledge. On the one hand, with unclear strategies, almost any knowledge activity could be seen as potentially useful to some extent for someone at some time or other. On the other hand, the broad set of missions makes it difficult to systematically set priorities among different kinds of knowledge activities. Major progress in evaluating knowledge activities will depend in part on clarifying the mission.
13. **Staff expertise:** Does the organization's staff have the deep expertise needed to tackle its strategy? In some cases, systematic de-skilling will have depleted needed expertise. The organization will need to consider whether such staff provide the organization with the necessary depth and continuity of expertise and experience, and the management skills, needed for the strategy it is pursuing.
14. **Organizational culture shift:** Historically, most organizations have seen themselves as being the source of the knowledge which it imparted to its customers; most of the knowledge was seen as coming from the developed countries. As noted above, the locus of knowledge is now much more geographically dispersed, with some key expertise existing only in the so-called developing countries. Moreover the top-down command-and-control management practices still prevalent in organizations in developed countries look increasingly anachronistic in the 21st Century in comparison with more agile approaches that have emerged. A question will be whether the organization will be able to shift its organizational culture from one of imparting its own knowledge to one of enabling access to knowledge wherever it may be located.
15. **Measuring outcomes flowing from knowledge services:** Individual knowledge services can be evaluated in terms of the specific purposes envisaged. For

instance, if knowledge is aimed at improving lending operations, measures can be used to assess whether it is having the intended outcomes. If knowledge services are aimed at improving decisions by customers or stakeholders, measures can be used to assess whether they are leading to the intended outcomes. If knowledge services are aimed at generating public goods, measures can be used to assess whether and to what extent those public goods are having the intended outcomes.

Summary

To summarize our discussion about, knowledge is power, we quote Stephen Hawking, where he says, "**The greatest enemy of knowledge is not ignorance, it is the illusion of knowledge.**"

As well as we can state that reality is the illusion created by the mind, based on perceptions. Change the perception and reality changes. The reality is created through your belief and by opening your mind and expand your knowledge, you will grow your reality.

References

[1] Bahman Zohuri and Masoud Moghaddam, Business Resilience System (BRS): Driven Through Boolean, Fuzzy Logics and Cloud Computation: Real and Near Real Time Analysis and Decision Making System 1st ed. 2017 Edition.

[2] David Derbyshire for Mail Online http://www.dailymail.co.uk/sciencetech/article-1355892/Each-person-inundated-174-newspapers-worth-information-EVERY-DAY.html.

[3] Lisa Quast, 'Why Knowledge Management Is Important To The Success Of Your Company' http://www.forbes.com/sites/lisaquast/2012/08/20/why-knowledge-management-is-important-to-the-success-of-your-company/#414146855e1d.

[4] David Garvin," Learning in Action: A Guide to Putting the Learning Organization to Work," Harvard Business Review Press (March 25, 2003).

[5] Pamela Babcock, "Shedding Light on Knowledge Management," https://www.shrm.org/hr-today/news/hr-magazine/Pages/0504covstory.aspx.

Chapter 2

FUZZY LOGIC CONCEPT

The idea of fuzzy logic was first, advanced by Dr. Lotfi Zadeh of the University of California at Berkeley in the 1960s. Dr. Zadeh was working on the problem of computer understanding of natural language. Natural language (like most other activities in life and indeed the universe) is not easily, translated into the absolute terms of 0 and 1. (Whether everything is ultimately describable in binary terms is a philosophical question worth pursuing, but in practice much data we might want to feed a computer is in some state in between and so, frequently, are the results of computing.) It may help to see fuzzy logic as the way reasoning really works and binary or Boolean logic is simply a special case of it.

2.1. INTRODUCTION

Fuzzy Logic (FL) is an approach to computing based on “degrees of truth” rather than the usual “true or false” (1 or 0) Boolean logic on which the modern computer is based.

Fuzzy logic includes 0 and 1 as extreme cases of truth (or “the state of matters” or “fact”) but also includes the various states of truth in between so that, for example, the result of a comparison between two things could be not “tall” or “short” but” .38 of tallness.”

Fuzzy logic seems closer to the way our brains work. We aggregate data and form a number of partial truths, which we aggregate further into higher truths, which in turn, when certain thresholds are, exceeded, cause certain further results such as motor reaction. A similar kind of process is, used in neural networks, expert systems and other Artificial Intelligence (AI) applications. In one short statement, *Fuzzy Logic is a logic that is, centered on multi-tier defaming and looks at aggregation of data and information, which is partially TRUE or partially FALSE*.

Each of these cases is, described as follows:

1. Neural Networks

A neural network usually involves a large number of processors operating in parallel and arranged in tiers. The first tier receives the raw input information -- analogous to optic nerves in human visual processing. Each successive tier receives the output from the tier preceding it, rather than from the raw input -- in the same way neurons further from the optic nerve receive signals from those closer to it. The last tier produces the output of the system.

Each processing node has its own small sphere of knowledge, including what it has seen and any rules it was originally programmed with or developed for itself. The tiers are highly interconnected, which means each node in tier n will be, connected to many nodes in tier n-1 -- its inputs -- and in tier n+1, which provides input for those nodes. There may be one or multiple nodes in the output layer, from which the answer it produces can be, read.

Neural networks are notable for being adaptive, which means they modify themselves as they learn from initial training and subsequent runs provide more information about the world. The most basic learning model is, centered on weighting the input streams, which is how each node weights the importance of input from each of its predecessors. Inputs that contribute to getting right answers are weighted higher.

2. Expert Systems

Typically, an expert system incorporates a knowledge base containing accumulated experience and an inference or rules engine -- a set of rules for applying the knowledge base to each particular situation that is, described to the program. The system's capabilities can be, enhanced with additions to the knowledge base or to the set of rules. Current systems may include machine-learning capabilities that allow them to improve their performance based on experience, just as humans do.

The concept of expert systems was first, developed in the 1970s by Edward Feigenbaum, professor and founder of the Knowledge Systems Laboratory at Stanford University. Feigenbaum explained that the world was moving from data processing to “knowledge processing,” a transition, which was being enabled by new processor technology and computer architectures.

Expert systems have played a large role in many industries including in financial services, telecommunications, healthcare, customer service, transportation, video games, manufacturing, aviation and written communication. Two early expert systems broke ground in the healthcare space for medical diagnoses: Dendral, which helped chemists identify organic molecules, and MYCIN, which helped to identify bacteria such as bacteremia and meningitis, and to recommend antibiotics and dosages.

3. Artificial Intelligence

AI (pronounced AYE-EYE) or artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using the rules to reach approximate or definite conclusions), and self-correction. Particular applications of AI include:

- Expert systems,
- Speech recognition, and
- Machine vision

Fuzzy logic is essential to the development of human-like capabilities for AI, sometimes referred to as artificial general intelligence: the representation of generalized human cognitive abilities in software so that, faced with an unfamiliar task, the AI system could find a solution.

2.2. What Is Fuzzy Logic and How It Works

The Fuzzy Logic was first, introduced to the world, by Persian Professor Dr. Lotfi Zadeh in 1965 when he was developing the theory of Fuzzy Sets. While the Fuzzy Logic theory can be quite complex, the scope of this book is not to cover the complete Fuzzy Logic theory and functions since many books are written for that purpose.

Therefore, we have tried to shed some light on the aspects of Fuzzy Logic, which can be used for our specific purpose and to show the application of Fuzzy Logic for our specific purpose, which is the Artificial Intelligence (AI) system in support a very reliable Business Resilience Systems (BRS) [1].

So, let us see what Boolean Logic is first: This concept was defined in Chapter 6 of the book and it is summarized here again.

A computer system or more precisely the Arithmetic Logic Unit (ALU), which is part of the Central Processing Unit (CPU), uses Boolean Logic to perform all kinds of calculations and computations. To do that, electrical elements called “Registers” are, implemented inside ALU. Registers hold Data required performing processes and they have two parts:

1. Keys
2. Values

A key has a value of ONE if it is turned ON or a value of ZERO when it is turned off. These values represent TRUE or FALSE or YES or NO in human language once they are interpreted by the operating system and other application programs. Therefore Boolean Logic (0 or 1) values are transformed into meaningful expressions by various programs installed on a computer to make all these complex calculations more understandable for us.

So, everything in a computer system works based on logical and mathematical calculations (Boolean Logic) resulted from millions of comparisons between ZEROs and ONEs!

But in real world we cannot judge everything on a Black and White or Yes or No basis. The same way that we have different shades of grey between black and white, we have different values or degrees of agreement or disagreement between Yes or No and Zero or One.

That is where the Fuzzy Logic (FL) fills the gap and can be, used for more precise calculations and analysis.

A good example in real life for a Fuzzy Value is a device called Thermostat. Without Thermostat we can either turn the air condition On or Off for a while because it either becomes absolute cold or absolute warm! Thanks to Thermostat, we can have degrees of different temperatures set on our air conditioning system to provide us with the best desirable temperature.

The following chart depicted in Figure 2-1 shows how a Fuzzy Logic works on a temperature control system:

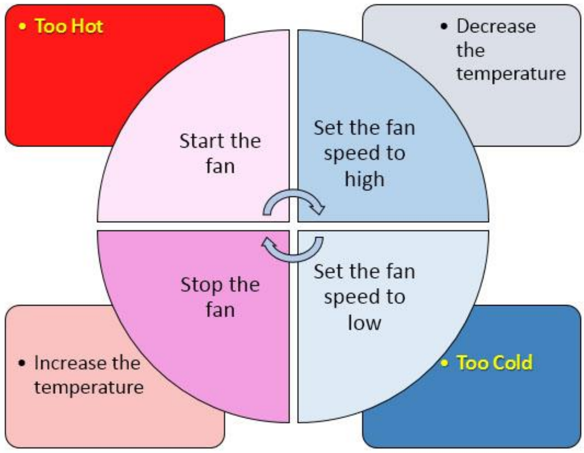


Figure 2-1. Illustration of Temperature Control System via Fuzzy Logic.

Now that have a simple introduction to the fuzzy logic and its usage in application of temperature control system, we can expand on it a little further on same application, by looking at FL in a role as an AI. Fuzzy Logics for most of us is not as fuzzy as you might think and has been working quietly behind the scenes for years. Fuzzy logic is a rule-based system that can rely on the practical experience of an operator, particularly useful to capture experienced operator knowledge. Here is what you need to know.

Fuzzy logic has been working quietly behind the scenes for more than 20 years in more places than most admit. Fuzzy logic is a rule-based system that can rely on the practical experience of an operator, particularly useful to capture experienced operator knowledge. Fuzzy logic is a form of artificial intelligence software; therefore, it would be, considered a subset of AI. Since it is performing a form of decision making, it can be loosely included as a member of the AI software toolkit. Here is what you need to know to consider using fuzzy logic to help solve your next application. It is not as fuzzy as you might think.

Fuzzy logic has been around since the mid 1960s; however, it was not until the 70s that a practical application was, demonstrated. Since that time the Japanese have traditionally been the largest producer of fuzzy logic applications. Fuzzy logic has appeared in cameras, washing machines, and even in stock trading applications. In the last decade, the United States has started to catch on to the use of fuzzy logic. There are many, applications that use fuzzy logic, but fail to tell us of its use. Probably the biggest reason is that the term “fuzzy logic” may have a negative connotation.

Fuzzy logic can be applied to non-engineering applications as illustrated in the stock trading application. It has also been used in medical diagnosis systems and in handwriting recognition applications. In fact a fuzzy logic system can be applied to almost any type of system that has inputs and outputs.

Fuzzy logic systems are well suited to nonlinear systems and systems that have multiple inputs and multiple outputs. Any reasonable number of inputs and outputs can be accommodated. Fuzzy logic also works well when the system cannot be modeled easily by conventional means.

Many engineers are afraid to dive into fuzzy logic due to a lack of understanding. Fuzzy logic does not have to be hard to understand, even though the math behind it can be intimidating, especially to those of us who have not been in a math class for many years.

Binary logic is either 1 or 0. Fuzzy logic is a continuum of values between 0 and 1. This may also be thought of as 0% to 100%. An example is the variable YOUNG. We may say that age 5 is 100% YOUNG, 18 is 50% YOUNG, and 30 is 0% YOUNG. In the binary world everything below 18 would be 100% YOUNG, and everything above would be 0% YOUNG.

The design of a fuzzy logic system starts with a set of membership functions for each input and a set for each output. A set of rules is then, applied to the membership functions to yield a "crisp" output value.

For this process control explanation of fuzzy logic, TEMPERATURE is the input and FAN SPEED is the output. Create a set of membership functions for each input. A membership function is simply a graphical representation of the fuzzy variable sets. For this example, use three fuzzy sets, COLD, WARM, and HOT. We will then create a membership function for each of three sets of temperature as shown in the cold-normal-hot graphic, Figure 2-2.

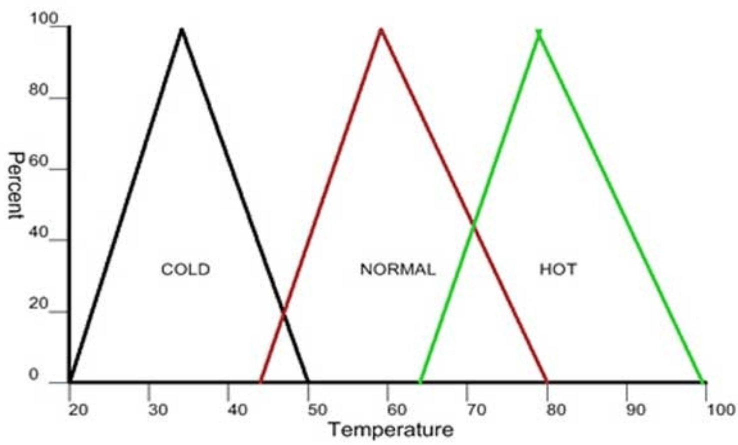


Figure 2-2. Illustration of Three Sets of Temperatures.

We will use three fuzzy sets for the output, SLOW, MEDIUM, and FAST. A set of functions is created for each output set just as for the input sets. It should be, noted that the shape of the membership functions do not need to be triangles as we have used in Figure 2-2 and Figure 2-3. Various shapes that can be used, such as Trapezoid, Gaussian, Sigmoid, as well as user definable. By changing the shape of the membership function, the user can tune the system to provide optimum response.

Now that we have our membership functions defined, we can create the rules that will define how the membership functions will be, applied to the final system. We will create three rules for this system.

- If HOT then FAST
- If WARM then MEDIUM
- If COLD then SLOW

The rules are then; applied to the membership functions to produce the "crisp" output value to drive the system. For simplicity, we will illustrate using only two input and two output functions.

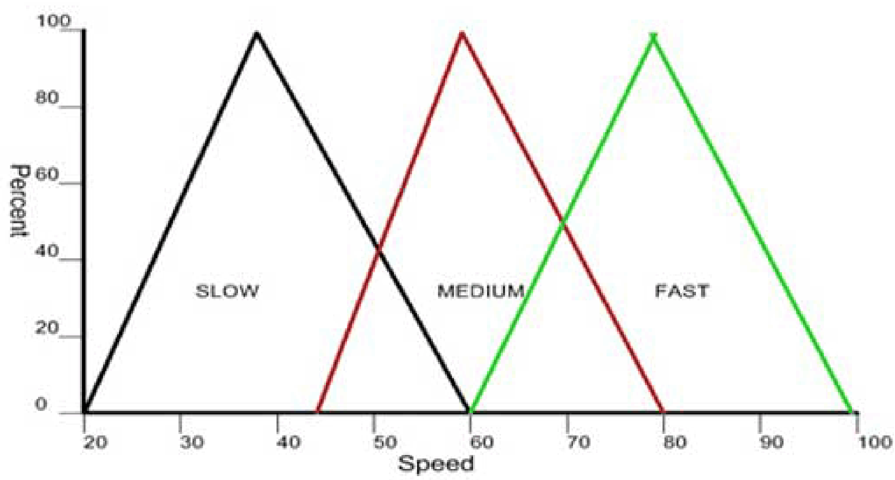


Figure 2-3. Different form of Three Sets of Temperatures Illustrations.

For an input value of 52 degrees, we intersect the membership functions. We see that in this example the intersection will be on both functions, thus two rules are applied. The intersection points are, extended to the output functions to produce an intersecting point. The output functions are then, truncated at the height of the intersecting points. The area under the curves for each membership function is then, added to give us a total area. The centroid of this area is calculated. The output value is then the centroid value. In this example, 44% is the output FAN SPEED value. This process is illustrated in Figure 2-4.

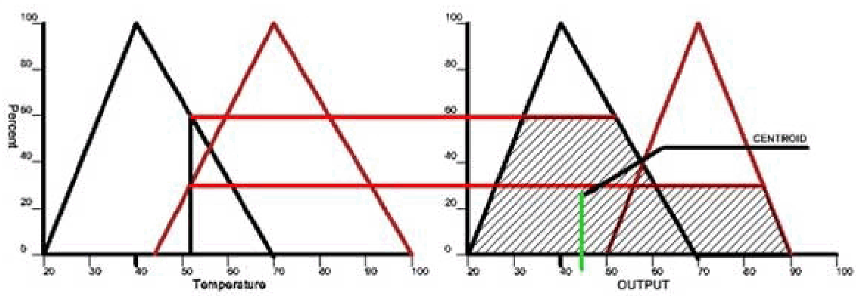


Figure 2-4. Illustration of Three Sets of Temperatures Processes.

This is a very simple explanation of how the fuzzy logic systems work. In a real working system, there would be many inputs and possibility several outputs. This would result in a fairly complex set of functions and many more rules. It is not uncommon for there to be 40 or more rules in a system. Even so, the principles remain the same as in our simple system.

National Instruments has included in LabVIEW© a set of pallet functions and a fuzzy system designer to greatly, simplify the task of building a fuzzy logic system. It has included

several demo programs in the examples to get started. In the graphical environment, the user can easily see what the effects are as the functions and rules are, built and changed.

The user should remember that a fuzzy logic system is not a “silver bullet” for all control system needs. Traditional control methods are still very much a viable solution. In fact, they may be, combined with fuzzy logic to produce a dynamically changing system. The validation of a fuzzy logic system can be difficult due to the fact, that it is a non-formal system. Its use in safety systems should be, considered with care.

We hope this short description will inspire the exploration and use of fuzzy logic in some of your future designs. We encourage the reader to do further study on the subject. There are numerous, books and articles that go into much more detail. This serves as a simple introduction to Fuzzy Logic Controls (FLC).

2.3. Fuzzy Logic and Fuzzy Sets

Fuzzy mathematics involves in general three operations as follow:

1. Fuzzyfication: Translation from real world values to Fuzzy values

It makes the translation from real world values to Fuzzy world values using membership functions. The membership functions in Figure 2-5, translate a speed= 55 into fuzzy values (Degree of membership) SLOW=0.25, MEDIUM=0.75 and FAST=0.

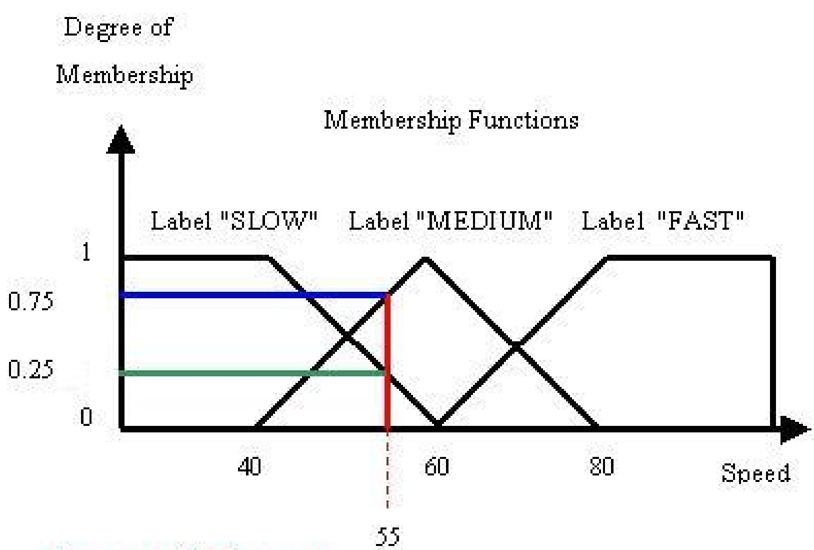


Figure 2-5. Illustration of Fuzzyfication.

2. Rule Evaluation: Computing rule strengths based on rules and inputs

Suppose SLOW=0.25 and FAR=0.82. The rule strength will be 0.25 (The minimum value of the antecedents) and the fuzzy variable INCREASE would be also 0.25.

Consider now another rule: If SPEED=MEDIUM and HIGHER=SECURE then GAS=INCREASE

Let be in this case, MEDIUM=0.75 and SECURE=0.5. Now the rule strength will be 0.5 (The minimum value of the antecedents) and the fuzzy variable INCREASE would be also 0.5.

Therefore, we have two rules involving fuzzy variable INCREASE. The “Fuzzy OR” of the two rules will be 0.5 (The maximum value between the two proposed values).

INCREASE=0.5

3. Defuzzyfication: Translate results back to the real world values

After compute the fuzzy rules and evaluate the fuzzy variables, we will need to translate these results back to the real world. We need now a membership function for each output variable like in Figure 2-6.

Let the fuzzy variables be:
DECREASE=0.2
SUSTAIN=0.8
INCREASE=0.5

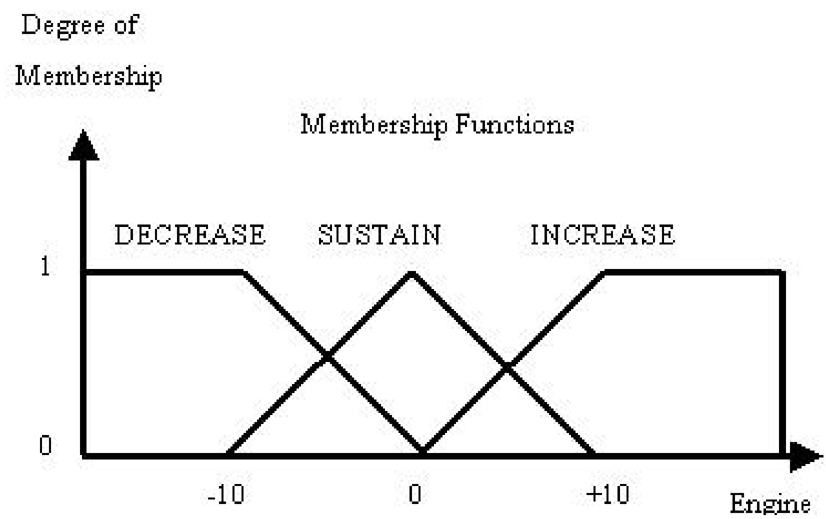


Figure 2-6. Illustration of Defuzzyfication.

Each membership function will be clipped to the value of the correspondent fuzzy variable as shown in Figure 2-7.

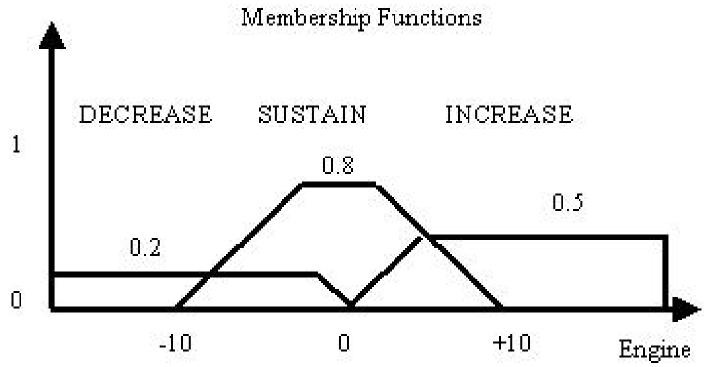


Figure 2-7. Illustration of Correspondent Fuzzy Variable.

A new output membership function is built, taking for each point in the horizontal axis, the maximum value between the three membership values. The result is shown in Figure 2-8.

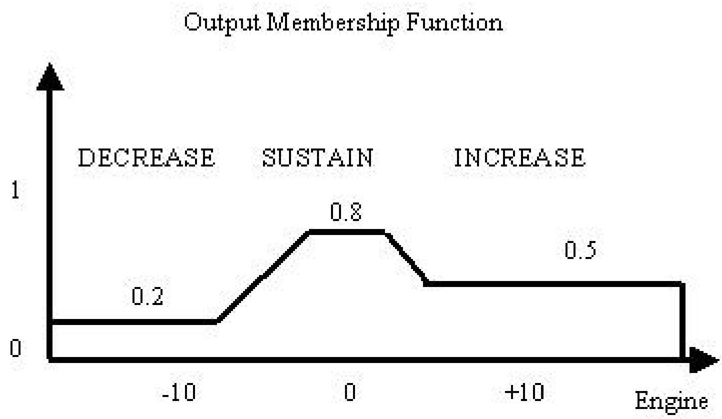


Figure 2-8. Result of Three Membership Values.

To complete the Defuzzyfication process, all we have to do now is found an equilibrium point. One way to do this is with "Center of Gravity (COG)" method.

$$COG = \frac{\int_a^b F(x) \cdot x dx}{\int_a^b F(x) dx} \qquad \text{Eq. 2-1}$$

This in our example will give us approximately, the following COG

$$COG \approx 2.6 \qquad \text{Eq. 2-2}$$

In summary, the Fuzzy Logic and Sets provide the following conclusions as:

- This theory lets us handle and process information in a similar way as the human brain does.
- We communicate and coordinate actions with data like "…you are too young to do that…" How much does "too" refer to, what's "young"?
- With fuzzy sets, we may define sub-sets in a fashion that any element may be part of them in different degrees.
- With fuzzy rules it's possible to compute relationships between fuzzy variables and produce fuzzy outputs.
- Moreover, guess what… from these fuzzy output values; we may build Boolean and continuous quantities, like a switch status or an amount of money.

Moreover, following is the base on which fuzzy logic is built:

As the complexity of a system increases, it becomes more difficult and eventually impossible to make a precise statement about its behavior, eventually arriving at a point of complexity where the fuzzy logic method born in humans is the only way to get at the problem.

(Originally identified and set forth by Lotfi A. Zadeh, Ph.D., University of California, Berkeley)

Fuzzy logic is used in system control and analysis design, because it shortens the time for engineering development and sometimes, in the case of highly complex systems, is the only way to solve the problem.

The following chapters of this book attempt to explain for us "Just Plain Folks" how the "fuzzy logic method born in humans" is used to evaluate and control complex systems. Although most of the time we think of "control" as having to do with controlling a physical system, there is no such limitation in the concept as initially presented by Dr. Zadeh. Fuzzy logic can apply also to economics, psychology, marketing, weather forecasting, biology, politics ...... to any large complex system.

The term "fuzzy" was first used by Dr. Lotfi Zadeh in the engineering journal, "Proceedings of the IRE," a leading engineering journal, in 1962. Dr. Zadeh became, in 1963, the Chairman of the Electrical Engineering department of the University of California at Berkeley. That is about as high as you can go in the electrical engineering field. Dr. Zadeh's thoughts are not to be taken lightly.

Fuzzy logic is not the wave of the future. It is now! There are already hundreds of millions of dollars of successful, fuzzy logic based commercial products, everything from self-focusing cameras to washing machines that adjust themselves according to how dirty the clothes are, automobile engine controls, anti-lock braking systems, color film developing systems, subway control systems and computer programs trading successfully in the financial markets.

Note that when you go searching for fuzzy-logic applications in the United States, it is difficult to impossible to find a control system acknowledged as based on fuzzy logic. Just imagine the impact on sales if General Motors announced their anti-lock braking was, accomplished with fuzzy logic! The general public is not ready for such an announcement maybe.

It should be, noted there is controversy and criticism regarding fuzzy logic. One must read various sides of the controversy and reach their own conclusion. Personally, the author, who has been both praised and reviled for his writings regarding fuzzy logic, feels the critics are too rigid in their grasp of the universe and "just don't get it." Nevertheless, do not take my word for it. You must look at all sides and make up your own mind.

2.4. THE FUZZY LOGIC METHOD

The fuzzy logic analysis and control method is, therefore:

1. Receiving of one, or a large number, of measurement or other assessment of conditions existing in some system we wish to analyze or control.
2. Processing all these inputs according to human based, fuzzy "If-Then" rules, which can be expressed in plain language words, in combination with traditional non-fuzzy processing.
3. Averaging and weighting the resulting outputs from all the individual rules into one single output decision or signal, which decides what to do or tells a controlled system what to do. The output signal eventually arrived at is a precise appearing, defuzzified, "crisp" value. Please see the following Fuzzy Logic Control/Analysis Method diagram as it shown in Figure 2-9:

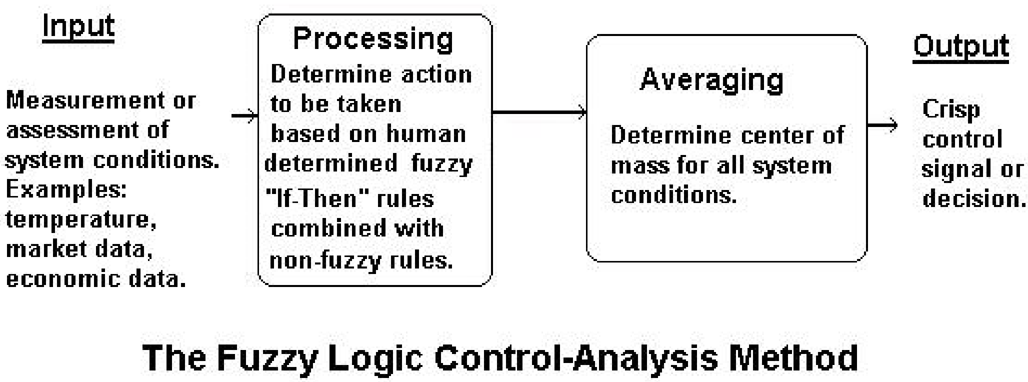


Figure 2-9. The Fuzzy Logic Control Analysis Method.

2.4.1. Fuzzy Perception

A fuzzy perception is an assessment of a physical condition that is not measured with precision, but is assigned an intuitive value. In fact, the fuzzy logic people assert everything in the universe is a little fuzzy, no matter how good your measuring equipment is. It will be seen below that fuzzy perceptions can serve as a basis for processing and analysis in a fuzzy logic control system.

Measured, non-fuzzy data is the primary input for the fuzzy logic method. Examples: temperature measured by a temperature transducer, motor speed, economic data, financial

markets data, etc. It would not be usual in an electro-mechanical control system or a financial or economic analysis system, but humans with their fuzzy perceptions could also provide input. There could be a human “in-the-loop.”

In the fuzzy logic literature, you will see the term “fuzzy set.” A fuzzy set is a group of anything that cannot be precisely, defined. Consider the fuzzy set of “old houses.” How old is an old house? Where is the dividing line between new houses and old houses? Is a fifteen-year-old house an old house? How about 40 years? What about 39.9 years? The assessment is in the eyes of the beholder.

Other examples of fuzzy sets are: tall women, short men, warm days, high pressure gas, small crowd, medium viscosity, hot shower water, etc.

When humans are the basis for an analysis, we must have a way to assign some rational value to intuitive assessments of individual elements of a fuzzy set. We must translate from human fuzziness to numbers that can be, used by a computer. We do this by assigning assessment of conditions a value from zero to 1.0. For “how hot the room is” the human might rate it at .2 if the temperature were below freezing, and the human might rate the room at .9, or even 1.0, if it is a hot day in summer with the air conditioner off.

You can see these perceptions are fuzzy, just intuitive assessments, not precisely measured facts.

By making fuzzy evaluations, with zero, at the bottom of the scale and 1.0 at the top, we have a basis for analysis rules for the fuzzy logic method, and we can accomplish our analysis or control project. The results seem to turn out well for complex systems or systems where human experience is the only base from which to proceed, certainly better than doing nothing at all, which is where we would be if unwilling to proceed with fuzzy rules.

2.4.2. Novices Can Beat the Pro's

Novices using personal computers and the fuzzy logic method can beat Ph.D. mathematicians using formulas and conventional programmable logic controllers. Fuzzy logic makes use of human common sense. This common sense is either, applied from what seems reasonable, for a new system, or from experience, for a system that has previously had a human operator.

Here is an example of converting human experience for use in a control system: I read of an attempt to automate a cement manufacturing operation. Cement manufacturing is a lot more difficult than you would think. Through the centuries it has evolved with human “feel” being absolutely necessary. Engineers were not able to automate with conventional control. Eventually, they translated the human “feel” into lots and lots, of fuzzy logic “If-Then” rules based on human experience. Reasonable success was thereby, obtained in automating the plant.

Objects of fuzzy logic analysis and control may include: physical control, such as machine speed, or operating a cement plant; financial and economic decisions; psychological conditions; physiological conditions; safety conditions; security conditions; production improvement and much more.

This book will talk about fuzzy logic in control applications - controlling machines, physical conditions, processing plants, etc. It should be. noted that when Dr. Zadeh invented

fuzzy logic, it appears he had in mind applying fuzzy logic in many applications in addition to controlling machines, such as economics, politics, biology, etc.

Thank You Wozniak (Apple Computer), Jobs (Apple Computer), Gates (Microsoft) and Ed Roberts (the MITS, Altair entrepreneur) for the Personal Computer.

The availability of the fuzzy logic method to us “just plain folks” has been made possible by the availability of the personal computer. Without personal computers, it would be difficult to use fuzzy logic to control machines and production plants, or do other analyses. Without the speed and versatility of the personal computer, we would never undertake the laborious and time consuming tasks of fuzzy logic based analyses and we could not handle the complexity, speed requirement and endurance needed for machine control.

You can do far more with a simple fuzzy logic BASIC or C++ program in a personal computer running in conjunction with a low cost input/output controller than with a whole array of expensive, conventional, programmable logic controllers.

Programmable logic controllers have their place! They are simple, reliable and keep American industry operating where the application is relatively simple and on-off in nature.

For a more complicated system control application, an optimum solution may be patching things together with a personal computer and fuzzy logic rules, especially if the project is being, done by someone, who is not a professional, control systems engineer.

2.4.3. A Milestone Passed for Intelligent Life on Earth

If intelligent life has appeared anywhere in the universe, “they” are probably using fuzzy logic. It is a universal principle and concept. Becoming aware of, defining and starting to use fuzzy logic is an important moment in the development of an intelligent civilization. On earth, we have just arrived at that important moment. You need to know and begin using fuzzy logic.

2.4.4. Fuzzy Logic Terms Found in Books and Articles

The discussion so far does not adequately prepare us for reading and understanding most books and articles about fuzzy logic, because of the terminology used by sophisticated authors. Following are explanations of some terms, which should help in this regard. This terminology was, initially established by Dr. Zadeh when he originated the fuzzy logic concept.

Fuzzy - The degree of fuzziness of a system analysis rule can vary between being very precise, in which case we would not call it “fuzzy,” to being based on an opinion held by a human, which would be “fuzzy.” Being fuzzy or not fuzzy, therefore, has to do with the degree of precision of a system analysis rule.

A system analysis rule need not be, based on human fuzzy perception. For example, you could have a rule, “If the boiler pressure rises to a danger point of 600 Psi as measured by a pressure transducer, then turn everything off. That rule is not “fuzzy.”

Principle of Incompatibility (previously stated; repeated here) -

- As the complexity of a system increases, it becomes more difficult and eventually impossible to make a precise statement about its behavior, eventually arriving at a point of complexity where the fuzzy logic method born in humans is the only way to get at the problem.
- Fuzzy Sets - A fuzzy set is almost any condition for which we have words: short men, tall women, hot, cold, new buildings, accelerator setting, ripe bananas, high intelligence, speed, weight, spongy, etc., where the condition can be given a value between 0 and 1. Example: A woman is 6 feet, 3 inches tall. In my experience, I think she is one of the tallest women I have ever met, so I rate her height at .98. This line of reasoning can go on indefinitely rating a great number of things between 0 and 1.
- Degree of Membership - The degree of membership is the placement in the transition from 0 to 1 of conditions within a fuzzy set. If a particular building's placement on the scale is a rating of .7 in its position in newness among new buildings, then we say its degree of membership in new buildings is .7.
- In fuzzy logic method, control systems, degree of membership is used in the following way. A measurement of speed, for example, might be, found to have a degree of membership in “too fast of” .6 and a degree of membership in “no change needed” of .2. The system program would then calculate the center of mass between “too fast” and “no change needed” to determine feedback action to send to the input of the control system. This is discussed in more detail in subsequent chapters.
- Summarizing Information - Human processing of information is not based on two-valued, off-on, either-or logic. It is based on fuzzy perceptions, fuzzy truths, fuzzy inferences, etc., all resulting in an averaged, summarized, normalized output, which is given by the human a precise number or decision value, which he or she verbalizes, writes down or acts on. It is the goal of fuzzy logic control systems to also do this.
- The input may be large masses of data, but humans can handle it. The ability to manipulate fuzzy sets and the subsequent summarizing capability to arrive at an output we can act on is one of the greatest assets of the human brain. This characteristic is the big difference between humans and digital computers. Emulating this human ability is the challenge facing those who would create computer, based artificial intelligence. It is proving very, very difficult to program a computer to have human-like intelligence.
- Fuzzy Variable - Words like red, blue, etc., are fuzzy and can have many shades and tints. They are just human opinions, not based on precise measurement in angstroms. These words are fuzzy variables.
- If, for example, speed of a system is the attribute being, evaluated by fuzzy, “fuzzy” rules, then “speed” is a fuzzy variable.
- Linguistic Variable - Linguistic means relating to language, in our case plain language words.

- Speed is a fuzzy variable. Accelerator setting is a fuzzy variable. Examples of linguistic variables are: somewhat fast speed, very high speed, real slow speed, excessively high accelerator setting, accelerator setting about right, etc.
- A fuzzy variable becomes a linguistic variable when we modify it with descriptive words, such as somewhat fast, very high, real slow, etc.
- The main function of linguistic variables is to provide a means of working with the complex systems mentioned above as being too complex to handle by conventional mathematics and engineering formulas.
- Linguistic variables appear in control systems with feedback loop control and can be related to each other with conditional, "if-then" statements. Example: If the speed is too fast, then back off on the high accelerator setting.
- Universe of Discourse - Let us make women the object of our consideration. All the women everywhere would be the universe of women. If we choose to discourse about (talk about) women, then all the women everywhere would be our Universe of Discourse.
- Universe of Discourse then, is a way to say all the objects in the universe of a particular kind, usually designated by one word, that we happen to be talking about or working with in a fuzzy logic solution.
- A Universe of Discourse is made up of fuzzy sets. Example: The Universe of Discourse of women is made up of professional women, tall women, Asian women, short women, beautiful women, and on and on.
- Fuzzy Algorithm - An algorithm is a procedure, such as the steps in a computer program. A fuzzy algorithm, then, is a procedure, usually a computer program, made up of statements relating linguistic variables.

2.5. THE WORLD'S FIRST FUZZY LOGIC CONTROLLER

In England in 1973 at the University of London, a professor and student were trying to stabilize the speed of a small steam engine the student had built. They had a lot going for them, sophisticated equipment like a PDP-8 minicomputer and conventional digital control equipment. But, they could not control the engine as well as they wanted. Engine speed would either, overshoot the target speed and arrive at the target speed after a series of oscillations, or the speed control would be too sluggish, taking too long for the speed to arrive at the desired setting, as in Figure 2-10, below.

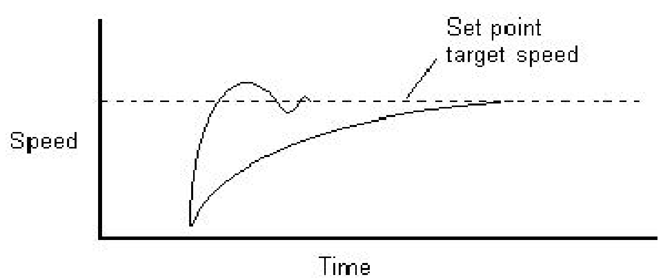


Figure 2-10. System Response without Fuzzy Logic Controller.

The professor, E. Mamdani, had read of a control method proposed by Dr. Lotfi Zadeh, head of the electrical engineering department at the University of California at Berkeley, in the United States. Dr. Zadeh is the originator of the designation “fuzzy,” which everyone suspects was selected to throw a little “pie in the face” of his more orthodox engineering colleagues, some of whom strongly opposed the fuzzy logic concept under any name.

Professor Mamdani and the student, S. Assilian, decided to try fuzzy logic. They spent a weekend setting their steam engine up with the world's first ever fuzzy logic control system and went directly into the history books by harnessing the power of a force in use by humans for 3 million years, but never before defined and used for the control of machines.

The controller worked right away, and worked better than anything they had done with any other method. The steam engine speed control graph using the fuzzy logic controller appeared as in Figure 2-11, below.

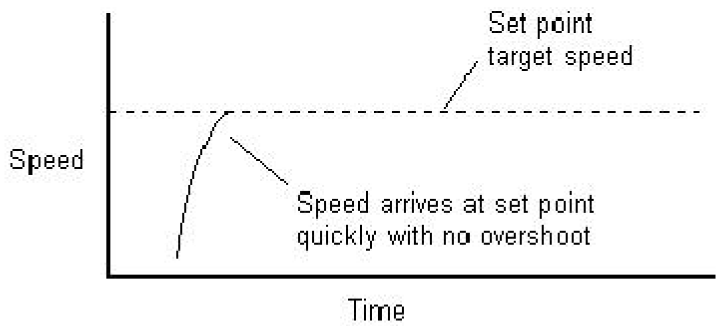


Figure 2-11. System Response Using Fuzzy Logic Controller [1].

As you can see, the speed approached the desired value very quickly, did not overshoot and remained stable. It was an exciting and important moment in the history of scientific development.

The Mamdani project made use of four inputs: boiler pressure error (how many temperature degrees away from the set point), rate of change of boiler pressure error, engine speed error and rate of change of engine speed error. There were two outputs: control of heat to the boiler and control of the throttle. They operated independently.

Note: A fuzzy logic system does not have to include a continuous feedback control loop as in the above, described Mamdani system in order to be a fuzzy-logic system, an impression you might receive from reading much of the fuzzy logic literature. There could be continuous feedback loop control, a combination of feedback loop control and on-off control or on-off control alone.

A fuzzy logic control system could be as simple as: “If the motor temperature feels like it is too hot, turn the motor off and leave it off.” Or, “If the company's president and all the directors just sold every share of stock they own, then WE sell!”

A fuzzy logic system does not have to be, directed toward electro-mechanical systems. The fuzzy logic system could be, for example, to provide buy-sell decisions to trade 30 million US dollars against the Japanese yen.

Fuzzy logic controllers can control solenoids, stepper motors, linear positioners, etc., as well as, or concurrently with, continuous feedback control loops. Where there is continuous feedback control of a control loop, the response for varying degrees of error can be non-

linear, tailoring the response to meet unique or experience determined system requirements, even anomalies.

Controllers typically have several inputs and outputs. The handling of various tasks, such as monitoring, and commanding a control loop, and monitoring various inputs, with commands issued as appropriate, would all be, sequenced in the computer program. The program would step from one task to the other, the program receiving inputs from and sending commands to the converter/controller.

Inputs for a fuzzy logic controlled mechanical/physical system could be, derived from any of thousands of real world, physical sensors/transducers. The Thomas Register has over 110 pages of these devices. Inputs for financial trading could come from personal assessments or from an ASCII data communication feed provided by a financial markets quote service.

2.5.1. Progress in Fuzzy Logic

From a slow beginning, fuzzy logic grew in applications and importance, until now it is a significant concept worldwide. Intelligent beings on the other side of our galaxy and throughout the universe have probably noted and defined the concept.

Personal computer based fuzzy logic control is pretty, amazing. It lets novices build control systems that work in places where even the best mathematicians and engineers, using conventional approaches to control, cannot define and solve the problem.

A control system is an electronic or mechanical system that causes the output of the controlled system to automatically, remain at some desired output (the “set point”) set by the operator. The thermostat on your air conditioner is a control system. Your car's cruise control is a control system. Control may be an on-off signal or a continuous feedback loop.

In Japan, a professor built a fuzzy logic control system that will fly a helicopter with one of the rotor blades off! Human helicopter pilots cannot do that. Moreover, the Japanese went further and built a fuzzy logic controlled subway that is as smooth as walking in your living room. You do not have to hang on to a strap to keep your balance. If you did not look out the window at things flashing by, you would hardly know you had started and were in motion.

In the United States, fuzzy logic control is gaining, popularity, but is not as widely, used as in Japan, where it is a multi-million dollar industry. Japan sells fuzzy logic controlled cameras, washing machines and more. One Internet search engine returns over 16,000 pages when you search on “fuzzy + logic.”

Personal computer based fuzzy logic control follows the pattern of human “fuzzy” activity. However, humans usually receive process and act on more inputs than the typical computer based fuzzy logic controller. (This is not necessarily so; a computer based fuzzy logic control system in Japan trades in the financial markets and utilizes 800 inputs) [2].

2.5.2. Fuzzy Logic Control Input - Human and Computer

Computer based fuzzy logic machine control is like human fuzzy logic control, but there is a difference when the nature of the computer's input is considered.

Humans evaluate input from their surroundings in a fuzzy manner, whereas machines/computers obtain precise appearing values, such as 112 degrees F, obtained with a

transducer and an analog to digital converter. The computer input would be the computer measuring, let us say, 112 degrees F. The human input would be a fuzzy feeling of being too warm.

The human says, “The shower water is too hot.” The computer as a result of analog input measurement says, “The shower water is 112 degrees F and 'If-Then' statements in my program tell me the water is too warm.” A human says, “I see two tall people and one short one.” The computer says, “I measure two people, 6' 6” and 6' 9”, respectively, and one person 5' 1” tall, and 'If-Then' statements in my program tell me there are two tall people and one short person.”

Even though transducer derived, measured inputs for computers appear to be more precise, from the point of input forward we still use them in a fuzzy logic method approach that follows our fuzzy, human approach to control.

For a human, if the shower water gets too warm, the valve handle is, turned to make the temperature go down a little. For a computer, an “If-Then” statement in the program would initiate the lowering of temperature based on a human provided “If-Then” rule, with a command output operating a valve [2].

2.5.3. More About How Fuzzy Logic Works

To create a personal computer based fuzzy logic control system, we:

1. Determine the inputs.
2. Describe the cause and effect action of the system with “fuzzy rules” stated in plain English words.
3. Write a computer program to act on the inputs and determine the output, considering each input separately. The rules become “If-Then” statements in the program. (As will be seen below, where feedback loop control is involved, use of graphical triangles can help visualize and compute this input-output action).
4. In the program, use a weighted average to merge the various actions called for by the individual inputs into one crisp output acting on the controlled system. (In the event there is only one output, then merging is not necessary, only scaling the output as needed).

The fuzzy logic approach makes it easier to conceptualize and implement control systems. The process is, reduced to a set of visualizable steps. This is a very important point. Actually implementing a control system, even a simple control system, is more difficult than it appears. Unexpected aberrations and physical anomalies inevitably occur. Getting the process working correctly ends up being a “cut and try” effort.

Experienced, professional digital control engineers using conventional control might know how to proceed to fine-tune a system. However, it can be difficult for us just plain folks. Fuzzy logic control makes it easier to visualize and set up a system and proceed through the cut and try process. It is only necessary to change a few plain English rules resulting in changing a few numbers in the program.

In reading about fuzzy logic control applications in industry, one of the significant points that stand out is: fuzzy logic is used because it shortens the time for engineering development.

Fuzzy logic enables engineers to configure systems quickly without extensive experimentation and to make use of information from expert human operators who have been performing the task manually.

Perhaps your control need is something a lot more down to earth than flying helicopters or running subways. Maybe all you want to do is keep your small businesses sawmill running smoothly, with the wood changing and the blade sharpness changing. Or, perhaps you operate a natural gas compressor for some stripper wells that are always coming on and going off, and you need to have the compressor automatically adjust in order to stay on line and keep the suction pressure low to get optimum production. Perhaps you dream of a race car that would automatically adjust to changing conditions, the setup remaining optimum as effectively as the above mentioned helicopter adjusts to being without a rotor blade.

There are a million stories, and we cannot guess what yours is, but chances are, if there is something you want to control, and you are not an experienced, full time, professional control engineer financed by a multi-million dollar corporation, then fuzzy logic may be for you. If you are all those things, it still may be for you.

A conventional programmable logic controller monitors the process variable (the pressure, temperature, speed, etc., that we want to control). If it is too high, a decrease signal is sent out. If it is too low, an increase signal is sent out. This is effective up to a point. But, consider how much more effective a control system would be if we use a computer to calculate the rate of change of the process variable in addition to how far away it is from the set point. If the control system acts on both these inputs, we have a better control system. In addition, that could be just the beginning; we can have a large number of inputs all is being, analyzed according to common sense and experience rules for their contribution to the averaged crisp output controlling the system.

Further, whereas conventional control systems are usually smooth and linear in performance, we sometimes encounter aberrations or discontinuous conditions, something that does not make good scientific sense and cannot be predicted by a formula, but it's there. If this happens, the fuzzy logic method helps us visualize a solution, put the solution in words and translate to “If - Then” statements, thereby obtaining the desired result. That is a very difficult thing to do with conventional Programmable Logic Controllers (known as PLC's). PLC's are programmable, but are far more limited than the program control available from a very simple BASIC program in a personal computer [1].

Fuzzy logic control is not based on mathematical formulas. This is a good thing, because, as easy as it might seem, it is difficult to impossible to write formulas that do what nature does. This is why novices using fuzzy logic can beat Ph.D. mathematicians using formulas. Fuzzy logic control makes use of human common sense. This common sense is either, applied from what seems reasonable, for a new system, or from experience, for a system that has previously had a human operator.

Some of the greatest minds in the technical world try to explain to others why fuzzy logic works, and other great technical minds contend that fuzzy logic is a “cop out.” The experts really “go at” each other. However, for us just plain folks, the fact is fuzzy logic does work, seems to work better than many expensive and complicated systems and is understandable and affordable.

2.6. RATIONAL FOR FUZZY LOGIC

Fuzzy Logic is about, a mathematical technique for dealing with imprecise data, and problems that have many solutions rather than one. Although it is implemented in digital computers which ultimately make only yes-no decisions, fuzzy logic works with ranges of values, solving problems in a way that more resembles human logic.

Fuzzy logic is used for solving problems with expert systems and real-time systems that must react to an imperfect environment of highly variable, volatile or unpredictable conditions. It "smoothes the edges" so to speak, circumventing abrupt changes in operation that could result from relying on traditional either-or and all-or-nothing logic.

Fuzzy logic was conceived by Lotfi Zadeh, former chairman of the electrical engineering and computer science department at the University of California at Berkeley. In 1964, while contemplating how computers could be programmed for handwriting recognition, Zadeh expanded on traditional set theory by making membership in a set a matter of degree rather than a yes-no situation.

To finally, finish this chapter, we look at the rational for Fuzzy Logic that was, stated by father of Fuzzy Logic, Professor Lotfi Zadeh himself and these points are:

- In the evolution of science a time comes when alongside the brilliant successes of a theory, T, what become visible are classes of problems, which fall beyond the reach of **T**. At that point, the stage is set for a progression from **T** to **T*--a** generalization of **T** Among the many historical examples are the transitions from Newtonian mechanics to quantum mechanics; from linear system theory to nonlinear system theory; and from deterministic models to probabilistic models in economics and decision analysis.
- Fuzzy logic is a better approximation to reality.
- In this perspective, a fundamental point--a point, which is not as, yet widely recognized--is that there are many classes of problems, which cannot be, addressed by any theory, **T**, which is based on bivalent logic. The problem with bivalent logic is that it is in fundamental conflict with reality–a reality in which almost everything is a matter of degree.
- To address such problems what is needed is a logic for modes of reasoning, which are approximate rather than exact. ***This is what fuzzy logic is, aimed at***.

2.7. INFORMATION PROCESSING DRIVEN BY FUZZY LOGIC

Information technology is one of the most rapidly changing disciplines, especially with the fuzzy extension. Fuzzy databases have been studied in many works and papers but, in general, these works study some particular area and many works are theoretical works, with very few real applications.

Data mining and data processing have conceived as a new form of accessing databases, and is considered again as a top-priority research line in resent activity around neural networking and artificial intelligence driven by information processing through Fuzzy Logic

(FL) approach. This comes very fruitful, where it is also emphasized that data mining contains in its own essence the task of answering some kind of imprecise query.

The use of Fuzzy Logic to manage the imprecise and/or uncertain information in databases is even older that the aforementioned reports about it. It is a research line that is widely consolidated and has been developing for over 20 years. Traditionally, two categories of work lines have been considered:

a) Those dealing with the problems of flexible querying to database, in general, consider that the user expresses the query by using imprecise terms and the result is often a set of elements of the database affected to an accomplishment degree.
b) Those addressing the description of data models include imprecise and/or uncertain attributes, relationships, and structures represented by fuzzy sets and fuzzy logic.

Fundamentally, a fuzzy database is a database with fuzzy characteristics, particularly fuzzy attributes. These may be defined as attributes of an item, or object in a database that allow the storage of fuzzy information either imprecise or uncertain data. There are many forms of adding flexibility in fuzzy databases. The simplest technique is to add a fuzzy membership degree to each record, that is, and attribute in the range [0,1]. However, there are other kinds of databases allowing fuzzy values to be stored in fuzzy attributes, using fuzzy sets including fuzzy spatial data types, possibly distributions, or fuzzy degrees associated with some attributes and with different meanings such as membership degree, importance degree, fulfillment degree, etc.

Sometimes, the expression *fuzzy databases* is used for classical databases with fuzzy queries or with other fuzzy aspects, such as constraints. The research on fuzzy database has been developing for about 20 years and is concentrated mainly on the following six research line, which are listed below:

1. Fuzzy querying in classical databases
2. Fuzzy queries on fuzzy databases
3. Extension of classical data models in order to achieve fuzzy databases, such as fuzzy relational databases, fuzzy object-oriented database, etc.
4. Fuzzy conceptual modeling tools
5. Fuzzy data mining techniques
6. Applications of these advances in real databases

Although there is a little interest about the forth issue in above list due to the fact that in general, there is little interest in fuzzy conceptual issues and this subject has been studied in some other works in a very exhaustive manner.

Querying with imprecision, contrary to classical querying, allows users to implement fuzzy linguistic labels, also named linguistic terms and express their preferences to better qualify the data they wish to get. An example of a flexible query, also named in this context a fuzzy query, would be a list of the young employees working in a department with a big budget. This query contains the fuzzy liguistic labels *young* and *big budget*. These labels are words, in natural language, that express or identify a fuzzy set fixed or context dependent. Summarizing, fuzzy queries are useful to reflect the prefernces of the end user and to rank the solutions.

The ability to make fuzzy queries in classical databases is very useful because currently there are many classical databases. The second research line includes the first one, but we prefer to separate them because this second line finds new problem that must be studied, and because it must be framed in a concrete fuzzy database models third research line. The first two lines are summarized by Zadrozny et al. [3].

In summary, Fuzzy Logic (FL) is only a mathematical tool. It is possibly the best tool for treating uncertain, vague, or subjective information, based on definition of this logic, which is partially *False* or *True* in contrast to Boolean Logic (BL), which either *False* or *True* as a definite decision making points of references.

Just to give an idea about the importance of this soft computing tool, we can mention the big quantity of publications in this field, including two research journals of great quality:

- Fuzzy Sets and Systems, and
- IEEE Transactions on Fuzzy Systems

In particular, Fuzzy Logic (FL) has been applied to databases in many scientific papers and real applications. Undoubtedly, it is a modern research field and it has a long road ahead.

A fuzzy extension of the SQL query language, called the PFSQL, can be defined. An interpreter for that language is integrated inside an implementation of the fuzzy JDBC driver. An implementation of the CASE tool for modeling of fuzzy relational database schemas rounds up a set of tools for the implementation of Java fuzzy database applications [4].

In summary, since fuzzy logic inception by Professor Lotfi Zadeh 50 years ago, and rapid emergence in the following decades, fuzzy sets and fuzzy logic have found their way into numerous fields of application, such as engineering and control, operations research and optimization, databases and information retrieval, data analysis and statistics, just to name a few. More recently, fuzzy concepts have also been used in machine learning, giving birth to the field of *fuzzy machine learning*. This development has largely been triggered by the increasing popularity of machine learning as a key methodology of Artificial Intelligence (AI), modern information technology and the data sciences. Moreover, it has come along with a shift from *knowledge-based* to *data-driven* fuzzy modeling, i.e., from the manual design of fuzzy systems by human experts to the automatic construction of such systems by fitting fuzzy models to data.

In more classical applications like information processing and expert systems, fuzzy logic is primarily used for the purpose of knowledge representation, and inference is mostly of a *deductive nature*. Machine learning, on the other hand, is mainly concerned with *inductive inference*, namely, the induction of general, idealized models from specific, empirical data. Thus, while the key importance of probability theory and statistics as mathematical foundations of machine learning is immediately understandable and indisputable, the role of fuzzy logic in this field is arguably much less obvious at first sight.

2.8. FUZZY LOGIC SYSTEM TYPE-1 AND TYPE-2

As we have learned fuzzy reasoning and probability are related to each other as well as they compliment one to another. For example, if we have to define the probability of

appearance of an edge in few frames of images, we have to define, what is an edge. Certain threshold for rate of variation has to be taken, which may not be true for other images or noisy images. Fuzzy logic, unlike probability, handles imperfection in the informational content event [1].

Basically, Type-2 fuzzy sets and systems generalize Type-1 fuzzy sets and systems so that more uncertainty can be handled. From the very beginning of fuzzy sets, criticism was made about the fact that the membership function of a type-1 fuzzy set has no uncertainty associated with it, something that seems to contradict the word fuzzy, since that word has the connotation of lots of uncertainty. So, what does one do when there is uncertainty about the value of the membership function? The answer to this question was provided in 1975 by the inventor of fuzzy sets, Prof. Lotfi A. Zadeh [27], when he proposed more sophisticated kinds of fuzzy sets, the first of which he called a type-2 fuzzy set. A type-2 fuzzy set lets us incorporate uncertainty about the membership function into fuzzy set theory, and is a way to address the above criticism of type-1 fuzzy sets head-on. And, if there is no uncertainty, then a type-2 fuzzy set reduces to a type-1 fuzzy set, which is analogous to probability reducing to determinism when unpredictability vanishes.

The above can be summarized in two separate frameworks. Fundamentally, there are two frameworks for Fuzzy Systems (FS), and they are listed below as:

1. Development based on Crisp mathematical model and fuzzifying some quantities:

Model 1: Fuzzy Mathematical Model
Example: Fuzzy – K means clustering

2. Development based on Fuzzy Inference rules:

Model 2: Fuzzy Logical Model
Example: Fuzzy decision Support System

2.8.1. Definition of Fuzzy Set

At the beginning of this chapter we talked about Fuzzy Set (FS) and we summarized its concept here as by defining the membership function of fuzzy set.

1. Concept for Fuzzy Set

- Definition (Membership Function of Fuzzy Set)

In fuzzy sets, each elements is mapped to [0,1] by membership function.

$$\mu_A : X \to [0,1] \qquad \text{Eq. 2-3}$$

where [0,1] means real numbers between 0 and 1 including the numbers 0 and 1. Not that the symbol X is considered as Universal Set and μ_A is membership function A.

Examples of fuzzy set definition can be depicted in the following images as Figures 2-12a and 2-12b.

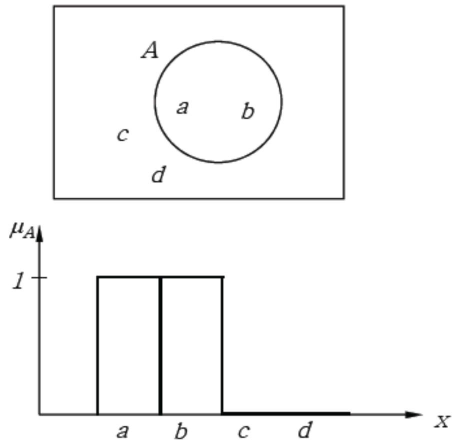


Figure 2-12(a). Graphical Representation of Crisp Set.

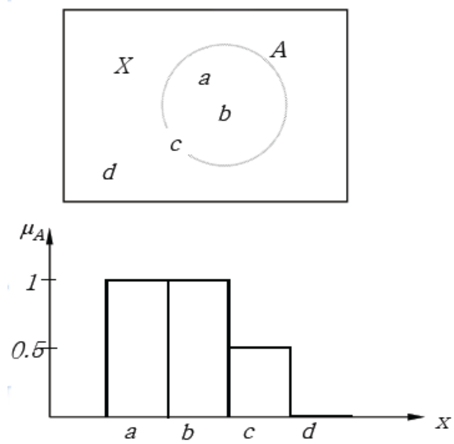


Figure 2-12(b). Graphical Representation of Fuzzy Set.

Now consider fuzzy set 'Two or so'. In this case universal set X are the positive real numbers as:

$$X = \{1, 2, 3, 4, 5, 6, \cdots\} \quad \text{Eq. 2-4}$$

Then membership function for A ='Two or so' in this universal set X is given as follows and presented by Figure 2-13:

$$\mu_A(1) = 0.5, \ \mu_A(2) = 1.0, \ \mu_A(3) = 0.5, \ \mu_A(1) = 0.0, \ \cdots \quad \text{Eq. 2-5}$$

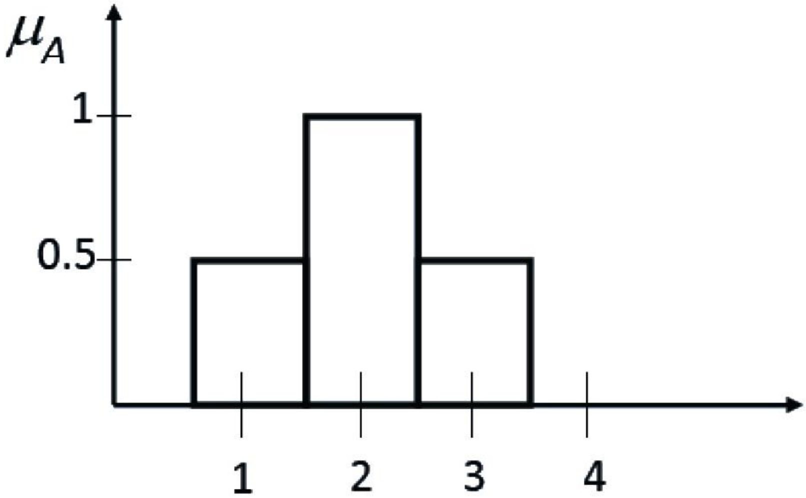


Figure 2-13. Depiction of Fuzzy Set 'Two or So.'

And if we want to present the linguistic terms of fuzzy set that was mentioned in as an example in and illustration format, Figure 2-14 is, then depicted.

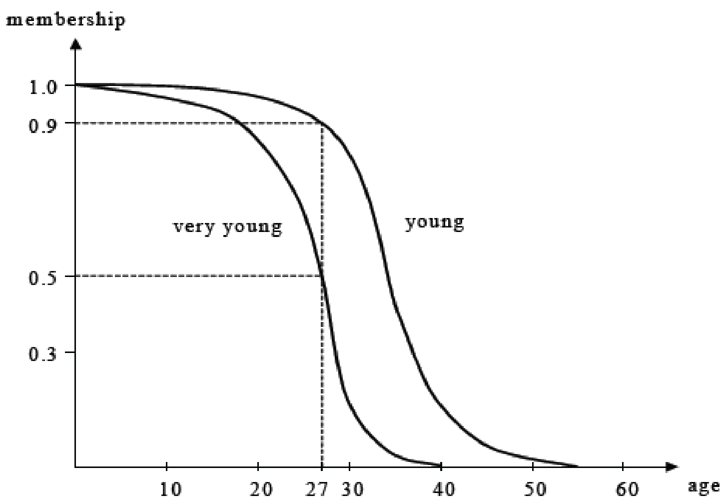


Figure 2-14. Fuzzy Sets Representing “Young” and “Very Young.”

Another examples of Fuzzy Set for membership function of real number near zero, both mathematically and as illustration are given as follows:

$$\begin{cases} A = \{\text{real number near } 0\} \\ A = \int \mu_A(x)/x \text{ where } \mu_A(X) = \dfrac{1}{1+x^2} \end{cases} \qquad \text{Eq. 2-6}$$

And illustration of it is:

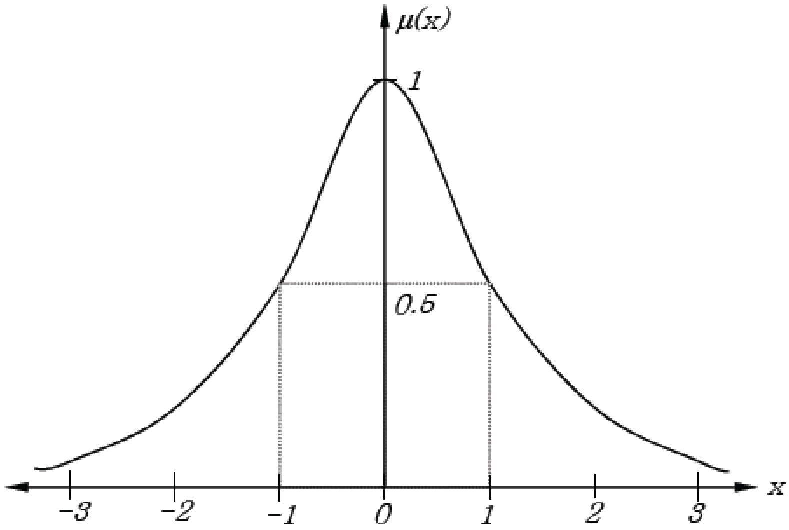


Figure 2-15. Membership Function of Fuzzy Set “Real Number Near 0.”

In order to symbolically distinguish between a type-1 fuzzy set and a type-2 fuzzy set, a tilde symbol is put over the symbol for the fuzzy set; so, A denotes a type-1 fuzzy set, whereas Ã denotes the comparable type-2 fuzzy set. When the latter is done, the resulting type-2 fuzzy set is called a general type-2 fuzzy set (to distinguish it from the special interval type-2 fuzzy set).

Prof. Zadeh didn't stop with type-2 fuzzy sets, because in that 1976 paper [5] he also generalized all of this to type-n fuzzy sets. The present article focuses only on type-2 fuzzy sets because they are the next step in the logical progression from type-1 to type-n fuzzy sets, where n = 1, 2, … . Although some researchers are beginning to explore higher than type-2 fuzzy sets, as of early 2009, this work is in its infancy.

The membership function of a general type-2 fuzzy set, $\tilde{A}$, is three-dimensional (Figure 2-16), where the third dimension is the value of the membership function at each point on its two-dimensional domain that is called its Footprint Of Uncertainty (FOU).

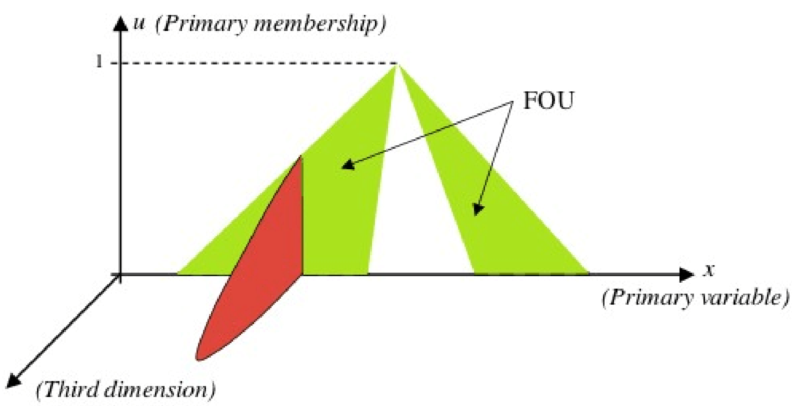


Figure 2-16. The Membership Function of a General Type-2 Fuzzy Set in Three-Dimensional Form.

Figure 2-16, is the membership of a general type-2 set in three-dimensional format. A cross-section of one slice of the third dimension is shown in reddish color in this figure. The cross-section, as well as all others, sits on the FOU. Only the boundary of the cross-section is used to describe the membership function of a general type-2 fuzzy set. It is shown filled-in for artistic purpose.

For an interval type-2 fuzzy set that third-dimension value is the same (e.g., 1) everywhere, which means that no new information is contained in the third dimension of an interval type-2 fuzzy set. So, for such a set, the third dimension is ignored, and only the FOU is used to describe it. It is for this reason that an interval type-2 fuzzy set is sometimes called a first-order uncertainty fuzzy set model, whereas a general type-2 fuzzy set (with its useful third-dimension) is sometimes referred to as a second-order uncertainty fuzzy set model.

The FOU represents the blurring of a type-1 membership function, and is completely described by its two bounding functions (Figure 2-17), a lower membership function (LMF) and an Upper Membership Function (UMF), both of which are type-1 fuzzy sets! Consequently, it is possible to use type-1 fuzzy set mathematics to characterize and work with interval type-2 fuzzy sets. This means that engineers and scientists who already know type-1 fuzzy sets will not have to invest a lot of time learning about general type-2 fuzzy set mathematics in order to understand and use interval type-2 fuzzy sets.

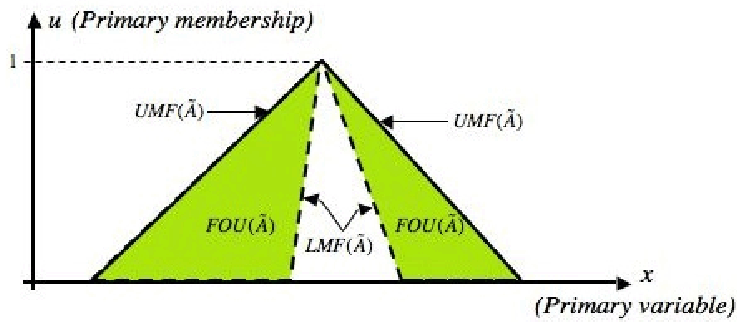


Figure 2-17. FOU for an Interval Type-2 Fuzzy Set.

Note that in Figure 2-17, many other shapes are possible for the Footprint of Uncertainty.

Work on type-2 fuzzy sets languished during the 1980s and early-to-mid 1990's, although a small number of articles were published about them. People were still trying to figure out what to do with type-1 fuzzy sets, so even though Zadeh proposed type-2 fuzzy sets in 1976, the time was not right for researchers to drop what they were doing with type-1 fuzzy sets to focus on type-2 fuzzy sets.

This changed in the latter part of the 1990s as a result of Prof. Jerry Mendel and his student's works on type-2 fuzzy sets and systems [6]. Since then, more and more researchers around the world are writing articles about type-2 fuzzy sets and systems.

Note that as well, the interval type-2 fuzzy sets have received the most attention because the mathematics that is needed for such sets—primarily Interval arithmetic—is much simpler than the mathematics that is needed for general type-2 fuzzy sets. So, the literature about interval type-2 fuzzy sets is large, whereas the literature about general type-2 fuzzy sets is much smaller. Both kinds of fuzzy sets are being actively researched by an ever-growing number of researchers around the world.

Now that, we have put some knowledge of Fuzzy Logic (FL) and Fuzzy Set (FS) as well as Fuzzy Logic System (FLS) under our belt, we can briefly talk about data-driven approach using the Type-2 fuzzy logic system by merging Type-1 fuzzy logic system [7].

Type-2Fuzzy Logic Systems (T2 FLSs) have shown superiorities in many real-world applications. With the exponential growth of data, it is a time consuming task to directly design a satisfactory T2 FLS through data-driven methods. There are research and presentation of an ensembling approach based data-driven method to construct T2 FLS through Type-1 Fuzzy Logic Systems (T1 FLSs), which are generated using the popular Adaptive Neuro Fuzzy Interface System (ANFIS) method.

2.9. Adaptive Neuro Fuzzy Interface System

Adaptive Neuro Fuzzy Interface System (ANFIS) are a class of adaptive networks that are functionally equivalent to fuzzy interface systems and represent ***Sugeno e Tsukamoto*** fuzzy model and implements a hybrid learning algorithm as well.

Sugeno Model, assumes that the fuzzy interface system has two inputs x and y and one out z.

A first-order Sugeno of Fuzzy model has rules as the following form:

- **Rule 1:**

  If x is A_1 and y is B_1, then $f_1 = p_1 x + q_1 y + r_1$

- **Rule 2:**

  If x is A_2 and y is B_2, then $f_2 = p_2 x + q_2 y + r_2$

Sugeno Model-I can be illustrated as follows

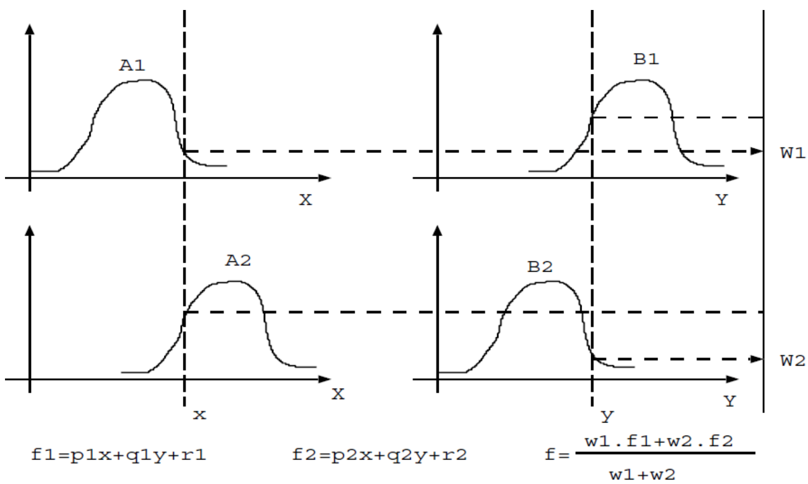


Figure 2-18. Sugeno Model-I.

ANFIS Architecture is depicted as Figure 2-19 in below form as well

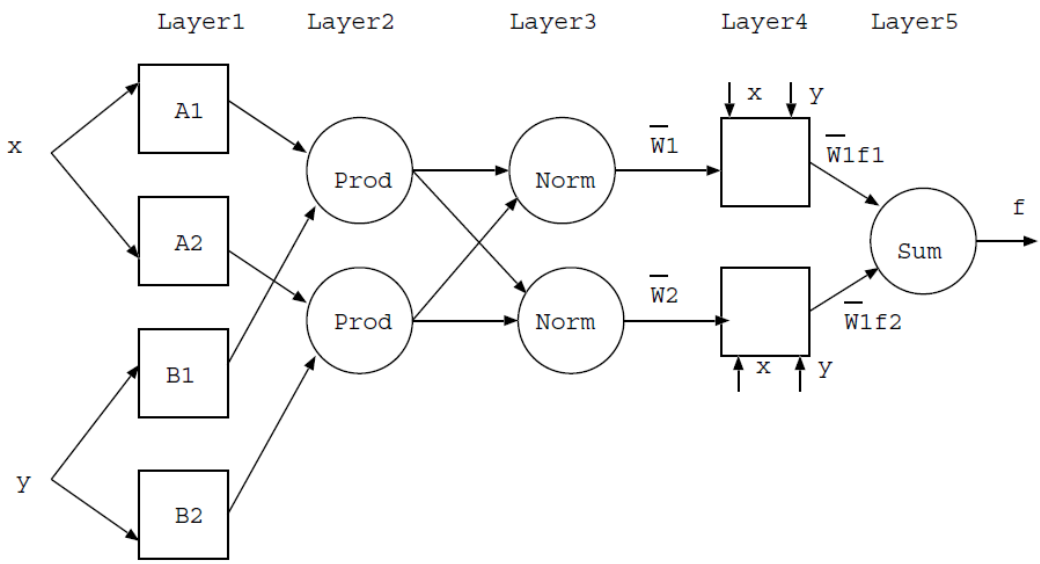


Figure 2-19. Illustration of ANFIS Architecture.

Adaptive Neuro Fuzzy Interface System (ANFIS) can be trained by a hybrid learning algorithm presented by Jang as part of learning Algorithm-I [8].

In summary, an adaptive Neuro Fuzzy Inference System or Adaptive Network-based Fuzzy Inference System (ANFIS) is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. The technique was developed in the early 1990s. Since it integrates both neural networks and fuzzy logic principles, it has potential to capture the benefits of both in a single framework. Its inference system corresponds to a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions. Hence, ANFIS is considered to be a universal estimator. For using the ANFIS in a more efficient and optimal way, one can use the best parameters obtained by genetic algorithm.

The constructed of T2 FLS could be applied to a wind speed prediction problem. Simulation and comparison of ANFIS, may show the results that compared with the well-known Back Propagation Neural Network (BPNN) as well.

2.10. Back Propagation Neural Network

The Back Propagation Neural Network (BPNN) that is illustrated in Figure 2-20, was developed by Rumelhart et al. [9] as a solution to the problem of training multilayer perceptrons. The fundamental advances represented by the BPNN were the inclusion of a differentiable transfer function at each node of the network and the use of error back-propagation to modify the internal network weights after each training epoch.

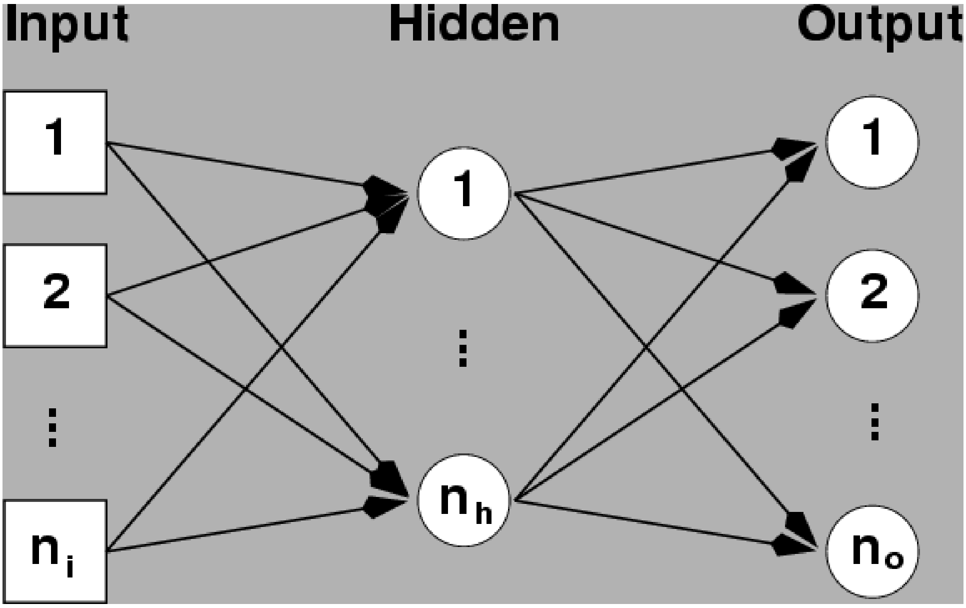


Figure 2-20. A Schematic of a Back Propagation Neural Network (BPNN).

The back-propagation neural networks used in this work all have three layers of neurons, or nodes (input, hidden, and output). Each node in the input and hidden layers is connected to each of the nodes in the next layer (hidden or output). All connections between nodes are directed (i.e., the information flows only one way), and there are no connections between the nodes within a particular layer. Each connection between nodes has a weighting factor associated with it. These weights are modified using the back-propagation algorithm during the training process to produce "learning."

The BPNN was chosen as a classifier primarily because of its ability to generate complex decision boundaries in the feature space [10]. There is even work suggesting that a BPNN, under appropriate circumstances, can approximate Bayesian posterior probabilities at its outputs [11]. This is significant because a Bayesian classifier provides the best performance possible (i.e., lowest error rate) for a given distribution of the feature data. As with other non-parametric approaches to pattern classification, it is not possible to predict the performance of a BPNN a priori. Furthermore, there are several parameters of the BPNN that must be chosen, including the number of training samples, the number of hidden nodes, and the learning rate.

Based on the work of Baum and Haussler [12], it is possible to place a bound (m) on the number of training samples needed to guarantee a particular level of performance on a set of test samples drawn from the same distribution as the training data. Specifically, if at least m samples are used to train a network with W weights and N nodes such that a fraction equal to $1-\frac{\varepsilon}{2}$ of them are classified correctly, then one can be confident that a fraction $1-\varepsilon$ of future (test) samples from the same distribution will be classified correctly, where

$$m \geq O\left(\frac{W}{\varepsilon}\log\frac{N}{\varepsilon}\right) \qquad \text{Eq. 2-7}$$

As a specific example, to guarantee no more than a 10% error in classifying the test data, the number of training samples should be equal to roughly 10 times the number of weights in the network. For a typical network generated below, this represents a requirement for 5000-10000 training samples. It is simply not tractable to generate that many images. Fortunately, this bound does not preclude the possibility of generating a successful classifier using fewer training samples, as many studies have empirically demonstrated.

The theoretical basis for selecting the number of hidden nodes to use in a single hidden layer network is not well developed. The only general method available to optimize this parameter is to test the network with various numbers of hidden nodes and select the one that performs best.

2.10.1. k-Nearest Neighbor Classifiers

Because a typical BPNN implementation has several parameters that must be chosen, a k-Nearest Neighbor (kNN) classifier (requiring the selection of a single parameter) was implemented to complement the BPNN results. One advantage of the kNN classifier is its intuitive operation. First, the distances between a single test sample and each of the training samples are calculated. The training samples closest to that test sample are defined as its "nearest neighbors." The test sample is then assigned to the class from which a plurality of its k nearest neighbors are from, where k is typically an integer less than 10.

In addition to having to optimize the selection of only a single parameter (k), the theoretical basis of the kNN classifier is well described [13]. The other major attraction of the kNN classifier is the fact that its asymptotic performance (as the number of training samples $\rightarrow \infty$ has been shown to be bounded by twice that of a Bayes classifier for the same data [14]. While the appeal of the kNN classifier lies in its simplicity and intuitive nature (i.e., new samples are assumed to belong to the same class as the training samples which are closest to them in the feature space), it is able to generate only piecewise-linear decision boundaries and is therefore not able to perform as well as the BPNN in practice.

2.11. Bayesian Network

A Bayesian Network (BN) or Bayes Network or sometimes called Belief Network model or probabilistic directed acyclic graphical model is a type of statistical model that represents a set of random variables and their conditional dependencies via a Directed Acyclic Graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms or information of data related to an events taking place. Given the symptoms or the trusted data and corrected information, the network can be used to compute the probabilities of the presence of various diseases in first case or probabilities an event taking place in second case.

Formally, Bayesian networks are DAGs whose nodes represent random variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; nodes that are not connected (there is no path from one of the variables to the other in the Bayesian network) represent variables

that are conditionally independent of each other. Each node is associated with a probability function that takes, as input, a particular set of values for the node's parent variables, and gives (as output) the probability (or probability distribution, if applicable) of the variable represented by the node. For example, if m parent nodes represent m Boolean variables then the probability function could be represented by a table of 2^m entries, one entry for each of the 2^m possible combinations of its parents being true or false. Similar ideas may be applied to undirected, and possibly cyclic, graphs; such as Markov networks.

Note that Boolean variables of Boolean type of data, in computer science, is the a data type, having two values that are usually denoted either as True or False and they intended to represent the *truth values* of *logic* and *Boolean algebra*. It is named after George Boole, who first defined an algebraic system of logic in the mid 19th century. The Boolean data type is primarily associated with conditional statements, which allow different actions and change control flow depending on whether a programmer-specified Boolean condition evaluates to true or false. It is a special case of a more general logical data type; logic does not always have to be Boolean.

Efficient algorithms exist that perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables (e.g., speech signals or protein sequences) are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams. An example of a simple Bayesian network node is depicted graphically in Figure 2-21. Rain influences whether the sprinkler is activated, and both rain and the sprinkler influence whether the grass is wet.

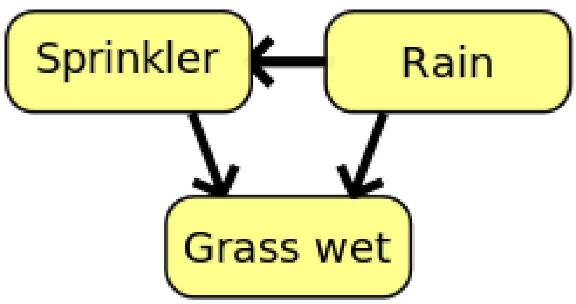


Figure 2-21. Simple Bayes Net Nodes.

Efficient algorithms exist that perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables (e.g., speech signals or protein sequences) are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

An example of Bayesian in a mathematical form is quoted here from Wikipedia site associated with depiction of Figure 2-21 in above and is expressed as follows.

Suppose that there are two events which could cause grass to be wet: either the sprinkler is on or it's raining. Also, suppose that the rain has a direct effect on the use of the sprinkler (namely that when it rains, the sprinkler is usually not turned on). Then the situation can be modeled with a Bayesian network (shown to the right). All three variables have two possible values, T (for true) and F (for false).

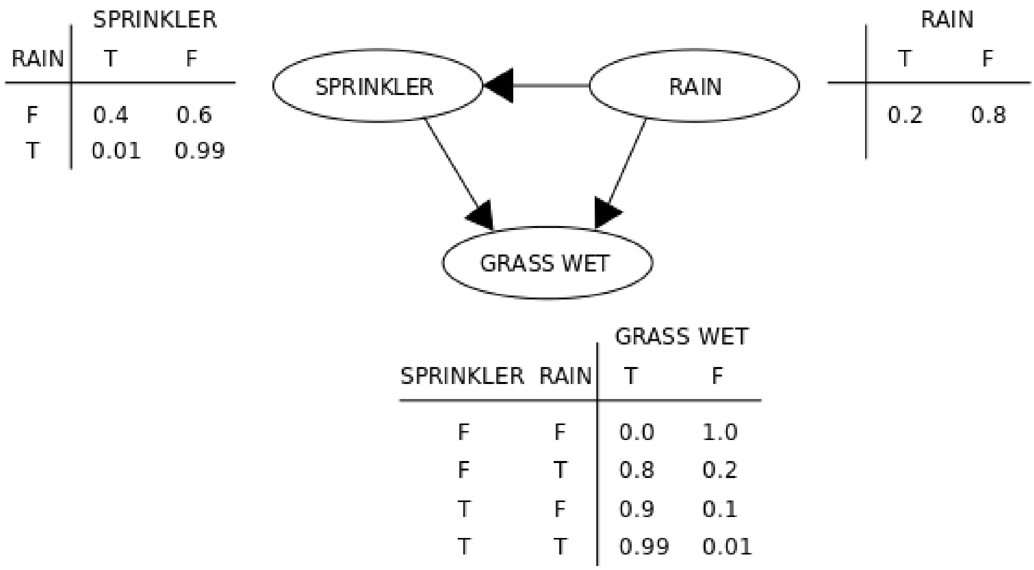


Figure 2-22. Simple Bayes Network with Conditional Probability Tables.

Utilizing Figure 2-22, the joint probability function is given as:

$$\Pr(G,S,R) = \Pr(G \mid S,R)\Pr(S \mid R)\Pr(R) \qquad \text{Eq. 2-8}$$

Where the names of the variables have been abbreviated to

G = Grass wet (Yes/No)
S = Sprinkler turned on (Yes/No), and Eq. 2-9
R = Raining (Yes/No)

The model can answer questions like "What is the probability that it is raining, given the grass is wet?" by using the conditional probability formula and summing over all nuisance variables, the we can write:

$$\Pr(R=T \mid G=T = \frac{\Pr(G=T,R=T)}{\Pr(G=T)} = \frac{\Sigma_{S\in(T,F)}\Pr(G=T,S,R=T)}{\Sigma_{S,R\in(T,F)}\Pr(G=T,S,R)} \qquad \text{Eq. 2-10}$$

Using the expansion for the joint probability function $\Pr(G,S,R)$ and the conditional probabilities from the Conditional Probability Tables (CPTs) stated in the Figure 2-22, one can evaluate each term in the sums in the numerator and denominator. For example:

$$\begin{aligned}\Pr(G=T,S=T,R=T) &= \Pr(G=T \mid S=T,R=T)\Pr(S=T \mid R)\Pr(R=T)\\ &= 0.99\times 0.01\times 0.2 \qquad \text{Eq. 2-11}\\ &= 0.00198\end{aligned}$$

Then the numerical results subscribed by the associated variable values are given as:

$$\Pr(R=T \mid G=T) = \frac{0.00198_{TTT} + 0.1584_{TFT}}{0.000198_{TTT} + 0.288_{TTF} + 0.1584_{TFT} + 0.0_{TFF}} = \frac{891}{2491} \approx 25.77\% \qquad \text{Eq. 2-12}$$

If, on the other hand, we wish to answer an interventional question and that is:

"What is the probability that it would rain, given that we wet the grass?"

The answer would be governed by the post-intervention joint distribution function as per Equation 2-13 below:

$$\Pr(S,R \mid \mathrm{do}(G=T)) = \Pr(S \mid R)\Pr(R) \qquad \text{Eq. 2-13}$$

If, furthermore, we wish to predict the impact of turning the sprinkler on, we can write

$$\Pr(S,R \mid \mathrm{do}(S=T)) = \Pr(R)\Pr(G \mid R, S=T) \qquad \text{Eq. 2-14}$$

with the term $\Pr(S=T \mid R)$ removed, showing that the action has an effect on the grass but not on the rain. These predictions may not be feasible when some of the variables are unobserved, as in most policy evaluation problems.

The effect of the action $\mathrm{do}(x)$ can still be predicted, however, whenever a criterion called "back-door" is satisfied. It states that, if a set Z of nodes can be observed that d-separates (or blocks) all back-door paths from X to Y then $\Pr(Y,Z \mid \mathrm{do}(x)) = \Pr(Y,Z,X=x) / \Pr(X=x \mid Z)$. A back-door path is one that ends with an arrow into X. Sets that satisfy the back-door criterion are called "sufficient" or "admissible." For example, the set $Z=R$ is admissible for predicting the effect of $S=T$ on G, because R d-separate the (only) back-door path $S \leftarrow R \rightarrow G$. However, if S is not observed, there is no other set that d-separates this path and the effect of turning the sprinkler on $(S=T)$ on the grass (G) cannot be predicted from passive observations. We then say that $\Pr(G \mid \mathrm{do}(S=T))$ is not "identified." This reflects the fact that, lacking interventional data, we cannot determine if the observed dependence between S and G is due to a causal connection or is spurious (apparent dependence arising from a common cause, R). (see Simpson's paradox)

To determine whether a causal relation is identified from an arbitrary Bayesian network with unobserved variables, one can use the three rules of "do-calculus" and test whether all do terms can be removed from the expression of that relation, thus confirming that the desired quantity is estimable from frequency data.

Simpson's Paradox

Simpson's paradox, or the Yule–Simpson effect, is a paradox in probability and statistics, in which a trend appears in different groups of data but disappears or reverses when these groups are combined. It is sometimes given the descriptive title reversal paradox or amalgamation paradox.

This result is often encountered in social-science and medical-science statistics, and is particularly confounding when frequency data is unduly given causal interpretations. The paradoxical elements disappear when causal relations are brought into consideration [5]. Many statisticians believe that the mainstream public should be informed of the counter-intuitive results in statistics such as Simpson's paradox. Martin Gardner wrote a popular account of Simpson's paradox in his March 1976 Mathematical Games column in Scientific American. Figure 2-23, illustrates, a Simpson's paradox for quantitative data: a positive trend (——, ——) appears for two separate groups, whereas a negative trend (......) appears when the groups are combined.

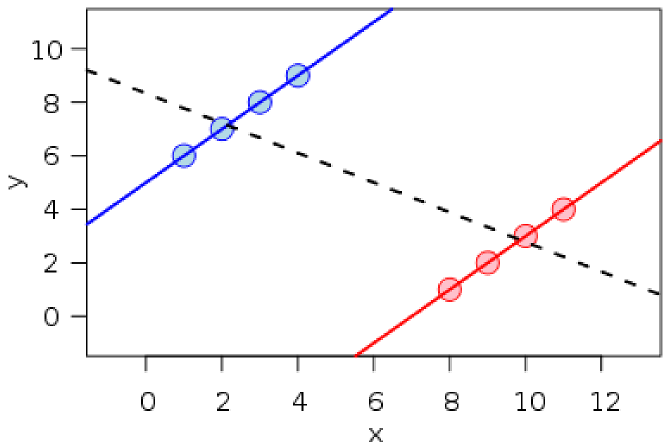


Figure 2-23. Simpson's Paradox for Quantitative Data.

Using a Bayesian network can save considerable amounts of memory, if the dependencies in the joint distribution are sparse. For example, a naive way of storing the conditional probabilities of 10 two-valued variables as a table requires storage space for 2^{10} = 1024 values. If the local distributions of no variable depends on more than three parent variables, the Bayesian network representation only needs to store at most 10 x 2^3 = 80 values.

One advantage of Bayesian networks is that it is intuitively easier for a human to understand (a sparse set of) direct dependencies and local distributions than complete joint distributions.

In summary an important feature of Bayesian networks is that they facilitate explicit encoding of information about independencies in the domain, information that is indispensable for efficient inferencing.

2.11.1. Bayesian Learning

As part artificial intelligence system based on foundation of computational agents, there exists a number of different learning algorithms and requires to cover some of the theoretical aspects of learning, developed in an area called computational learning theory, where readers can refer to the book by Poole and Mackworth [15]. Their approach is by raising some relevant questions that we can ask about a theory of computational learning including the following ones:

- Is the learner guaranteed to converge to the correct hypothesis as the number of examples increases?
- How many examples are required to identify a concept?
- How much computation is required to identify a concept?

In general, the answer to the first question is “no,” unless it can be guaranteed that the examples always eventually rule out all but the correct hypothesis. Someone out to trick the learner could choose examples that do not help discriminate correct hypotheses from incorrect hypotheses. So if such a person cannot be ruled out, a learner cannot guarantee to find a consistent hypothesis. However, given randomly chosen examples, a learner that always chooses a consistent hypothesis can get arbitrarily close to the correct concept. This requires a notion of closeness and a specification of what is a randomly chosen example.

Consider a learning algorithm that chooses a hypothesis consistent with all of the training examples. Assume a probability distribution over possible examples and that the training examples and the test examples are chosen from the same distribution. The distribution does not have to be known. Poole and Mackworth [15] prove a result that holds for all distributions and we recommend that readers refer to their book for further consulting on the first two questions.

To consider the third question, namely, how quickly a learner can find the probably approximately correct hypothesis. Again their attempt, first is, if the sample complexity is exponential in the size of some parameter (e.g., n in their examples), the computational complexity must be exponential because an algorithm must at least consider each example. To show an algorithm with polynomial complexity, we must find a hypothesis space with polynomial sample complexity and show that the algorithm uses polynomial time for each example. However, to go on moreover, they talk about Bayesian Learning, by expressing that, rather than choosing the most likely model or delineating the set of all models that are consistent with the training data, another approach is to compute the posterior probability of each model given the training examples.

The idea of Bayesian learning is to compute the posterior probability distribution of the target features of a new example conditioned on its input features and all of the training examples and they present that in their book [15]. In that case, the set of models may include structurally different models in addition to models that differ in the values of the parameters. One of the techniques of Bayesian learning is to make the parameters of the model explicit and to determine the distribution over the parameters.

In the simple case as part of structure learning, a Bayesian network is specified by an expert and is then used to perform inference. In other applications the task of defining the

network is too complex for humans. In this case the network structure and the parameters of the local distributions must be learned from data.

Automatically learning the graph structure of a Bayesian network is a challenge pursued within machine learning. The basic idea goes back to a recovery algorithm developed by Rebane and Pearl (1987) [16] and rests on the distinction between the three possible types of adjacent triplets allowed in a directed acyclic graph (DAG):

1. $X \rightarrow Y \rightarrow Z$
2. $X \leftarrow Y \rightarrow Z$
3. $X \rightarrow Y \leftarrow Z$

Type 1 and type 2 represent the same dependencies (X and Z are independent given Y) and are, therefore, indistinguishable. Type 3, however, can be uniquely identified, since X and Z are marginally independent and all other pairs are dependent. Thus, while the skeletons (the graphs stripped of arrows) of these three triplets are identical, the directionality of the arrows is partially identifiable. The same distinction applies when X and Z have common parents, except that one must first condition on those parents. Algorithms have been developed to systematically determine the skeleton of the underlying graph and, then, orient all arrows whose directionality is dictated by the conditional independencies observed.

An alternative method of structural learning uses optimization based search. It requires a scoring function and a search strategy. A common scoring function is posterior probability of the structure given the training data, like the Bayesian Information Criterion (BIC) or the Bayesian Dirichlet equivalent uniform (BDeu) score for learning Bayesian network (BN) structure. The time requirement of an exhaustive search returning a structure that maximizes the score is super-exponential in the number of variables. A local search strategy makes incremental changes aimed at improving the score of the structure. A global search algorithm like Markov chain Monte Carlo can avoid being, trapped in local minima. Friedman et al. [17] discuss using mutual information between variables and finding a structure that maximizes this. They do this by restricting the parent candidate set to k nodes and exhaustively searching therein.

Note that The BDeu score aims at maximizing the posterior probability of the DAG given data, while assuming a uniform prior over possible DAGs. In this work, we propose two new score functions, namely Min-BDeu and Max-BDeu. These scores are based on the BDeu score, but they consider all possible prior probability distributions inside an “-contaminated set [19] of Dirichlet priors around the symmetric one (which is the one used by the original BDeu).

A particularly fast method for exact BN learning is to cast the problem as an optimization problem, and solve it using Integer programming. Acyclicity constraints are, added to the Integer Program (IP) during solving in the form of cutting planes [18]. Such method can handle problems with up to 100 variables.

2.11.2. Application of Bayesian Networks

Businesses, Enterprises, organizations such as homeland security and intelligent agencies and governments must often asses and manage risk in areas where there is little or no direct historical data to draw upon, or where relevant data is difficult to identify. For example, the Barings Bank in England and some other banks in United States, collapse in 1995 (e.g., England) and 2006-2008 (e.g., USA) was not due to credit or market risk, where banks have sufficient data for prediction and mitigation of risk, but rather it was due to what is now called operational risk. This is the results of failures in everyday operational processes. The challenges are similarly acute when the source of the risk is novel: terrorist attacks, ecological disasters, major project failures, and more general failures of novel systems, market places and business models.

Even though we may have little or no historical data, there is often an abundance of expert (but subjective) judgment, as well as diverse information and data on indirectly related risks.

These are the types of situation that can be successfully addressed using Bayesian Networks (BNs), even when classical, data-driven approaches to risk assessment are not possible. BNs describe “webs” of causes and effects, using a graphical framework that provides for the rigorous quantification of risks and the clear communication of results. They can combine historical data with expert judgment [20].

During the last decade, researchers have incorporated BN techniques into easy-to-use toolsets, which in turn have enabled the development of decision support systems in a diverse set of application domains, including medical diagnosis, safety assessment, forensics, procurement, equipment fault diagnosis and software quality. Further technology and tool advancements since 2000 mean that end-users, rather than just researchers, are now able to develop and deploy their own BN-based solutions. As a result, BN methods are beginning to penetrate mainstream business practice. Recent commercial case studies provide evidence of impressive returns on investment from these techniques [20].

Both the practice and research of BNs are mushrooming. This report provides a snapshot of this dynamic and exciting area, including an introduction to the underpinning ideas, recent case studies, emerging areas of application, current research challenges, and a summary of the key players.

2.12. Fuzzy Logic Algorithms and Neural Networking

Fuzzy logic has evolved into a very useful tool for solving complex, real-world problems. Fuzzy logic is well suited to applications in linear and nonlinear control systems, signal and image processing, and other data analysis problems (Klir and Folger 1988 [21]; Kosko 1992 [22]). L. Zadeh did the pioneering work in this area beginning in 1965 (Yager et al. 1987; Klir and Yuan 1996). The strength of fuzzy logic algorithms lies in their ability to systematically, address the natural ambiguities in measurement data, classification, and pattern recognition. While fuzzy logic algorithms have been, successfully applied in the engineering sciences, the use of these techniques in any other sciences has been fairly, limited. The fuzzy logic techniques used in this work follow that of some literatures and couple books that exist in

open market. The reader should refer to those works for a more complete description of these techniques. Due to the inherent ambiguity in many aspects of neural networking and related artificial intelligence data measurement, analysis, detection, and forecasting algorithms, for decision making and building business resilience system based on fuzzy logic has the potential to become a very useful tool in this field.

There are varieties of ways that fuzzy logic can be, implemented, including fuzzy neural networks, fuzzy expert systems, and fuzzy inference systems. The current implementation is a variant of a fuzzy inference system and builds on two basic processes: *fuzzification* and *composition.* The first step, fuzzification, performs the conversion of measurement data into scaled, unitless numbers that indicate the correspondence or "membership level" of the data to the desired feature. The composition step combines the membership values from a number of different data types in a systematic fashion. The correspondence between a data value and the degree to which that data belongs to a certain class is, quantified by the application of a prescribed functional relation or *membership function.*

REFERENCES

[1] B. Zohuri and M. Moghaddam, Business Resilience System (BRS): Driven Through Boolean, Fuzzy Logics and Cloud Computation: Real and Near Real Time Analysis and Decision Making System 1st ed. 2017 Edition.

[2] Exis, LLC http://www.fuzzy-logic.com/.

[3] Jose Galindo, Handbook of Research on Fuzzy Information Processing in Databases 1st Edition, Published by Information Science Reference; (May 30, 2008).

[4] Srdjan Škrbić, Miloš Racković, Aleksandar Takači, Prioritized fuzzy logic based information processing in relational databases, Elsevier, *Knowledge-Based System*s, Volume 38, January 2013, Pages 62-73.

[5] L. A. Zadeh, "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning–1," *Information Sciences,* vol. 8, pp. 199–249, 1975.

[6] J. M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, Upper-Saddle River, NJ, 2001.

[7] Chengdong Li, Li Wang, Zixiang Ding, and Guiqing Zhang, School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan China.

[8] Jang, Sun, Mizutani (1997) – Neuro-Fuzzy and Soft Computing – Prentice Hall, pp 335–368.

[9] D. Rumelhart, G. Hinton, and Williams R., "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533-536, 1986.

[10] Kurt Hornick, Maxwell Stinchcombe, and Halbert White, "Multilayer feed forward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[11] Michael D. Richard and Richard P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation,* vol. 3, no. 4, pp. 461-483, 1991.

[12] E.B. Baum and D. Haussler, "What size net gives valid generalization?," *Neural Computation*, vol. 1, no. 1, pp. 151-160, 1989.

[13] Belur V. Dasarathy, Nearest neighbor (NN) norms: NN pattern classification techniques, IEEE Computer Society Press; *IEEE Computer Society Press Tutorial, Los Alamitos*, Calif. Washington, 1991.

[14] T.M. Cover and P.E. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. IT-13, pp. 21-27, 1967.

[15] David Poole and Alan Mackworth, “Artificial Intelligence, Foundation of Computational Agents,” Published by Cambridge University Press, 2010.

[16] Rebane, G. and Pearl, J., “The Recovery of Causal Poly-trees from Statistical Data,” Proceedings, 3rd Workshop on Uncertainty in AI, (Seattle, WA) pages 222–228, 1987.

[17] Friedman, Nir; Linial, Michal; Nachman, Iftach; Pe'er, Dana (August 2000). “Using Bayesian Networks to Analyze Expression Data.” *Journal of Computational Biology*. 7 (3-4): 601–620. doi:10.1089/106652700750050961. PMID 11108481. Retrieved 24 February 2015.

[18] Cussens, James (2011). “Bayesian network learning with cutting planes.” *Proceedings of the 27th Conference Annual Conference on Uncertainty in Artificial Intelligence:* 153–160.

[19] Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991).

[20] Norman Fenton and Martin Neil, “MANAGING RISK IN THE MODERN WORLD Applications of Bayesian Networks,” A Knowledge Transfer Report from the London Mathematical Society and the Knowledge Transfer Network for Industrial Mathematics, London Mathematical Society De Morgan House, 57/58 Russell Square London WC1B 4HS, November 2007.

[21] Klir, G. J., and T. A. Folger, 1988: *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, 355 pp.

[22] Kosko, B., 1992: *Neural Networks and Systems*: *A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, 449 pp.

Chapter 3

NEURAL NETWORK CONCEPT

Neural networks are a popular target representation for learning. These networks are inspired by the neurons in the brain but do not actually simulate neurons. Artificial neural networks typically contain many fewer than the approximately 10^{11} neurons that are in the human brain, and the artificial neurons, called units, are much simpler than their biological counterparts. Interconnectivity and interoperability between this concept as a foundation for Artificial Intelligence (AI) going forward is inevitable coupling. AIs of future are series of smart robots, that may get feeding of information from a central module to follow a new sets of Command, Control, Communication and Intelligence, known as C^3I, for their actions, while they could be independent and on their own, as much as their capability and functionality that built in them, allows.

3.1. INTRODUCTION

Artificial Intelligence of today are foundation of future smart robots to carry on whatever tasks could be assigned to them. These robots may be designed around the idea of processing data by far faster than human brain and to be considered as plat form for computational agents of near future going forward in time. Per description by (Poole and Mackworth) [1] "A computational agent is an agent whose decisions about its actions can be explained in terms of computation. That is, the decision can be broken down into primitive operation that can be implemented in a physical device. This computation can take many forms. In humans this computation is carried out in "***wetware,"*** in computers it is carried out in "***hardware***." Although there are some agents that are arguably not computational, such as the wind and rain eroding a landscape, it is an open question whether all intelligent agents are computational."

Additionally, Poole and Mackworth are clamming that an agent acts intelligently, when performs under the following conditions, where our interest falls into how this agent acts as well, so when can judge it by its action:

- What it does is appropriate for its circumstances and its goals,
- It is flexible to changing environments and changing goals,
- It learns from experience, and

- It makes appropriate choices given its perceptual and computational limitations. An agent typically cannot observe the state of the world directly; it has only a finite memory and it does not have unlimited time to act.

Thus the fundamental and infrastructure of a computational agent is an agent whose decisions about its actions can be explained in terms of computation. That is, the decision can be broken down into primitive operation that can be implemented in a physical device. The central *scientific goal* of Artificial Intelligence (AI) is to understand the principle that make intelligent behavior possible in natural or artificial systems. This is done by:

- The analysis of natural and artificial agents,
- Formulating and testing hypotheses about what it takes to construct intelligent agents, and
- Designing, building, and experimenting with computational systems that perform tasks commonly viewed as requiring intelligence

As part of science, researchers build empirical systems to test hypotheses or to explore the space of possibilities. These are quite distinct from applications that are built to be useful for an application domain.

It is arguable that intelligence is different: you cannot have fake intelligence. If an agent behaves intelligently, it is intelligent. It is only the external behavior that defines intelligence; acting intelligently is being intelligent. Thus, artificial intelligence, if and when it is achieved, will be real intelligence created artificially. However, an agent that is not really intelligent could not fake intelligence for arbitrary topics.

Note that: the definition is not for intelligent thought. We are only interested in thinking intelligently insofar as it leads to better performance. The role of thought is to affect action.

The central engineering goal of AI is the design and synthesis of useful, intelligent artifacts. We actually want to build agents that act intelligently. Such agents are useful in many applications [1].

Although, modern computers, from low-level hardware to high-level software, are more complicated than any human can understand, yet they are manufactured daily by organizations of humans. Human society viewed as an agent is arguably the most intelligent agent known.

3.2. Artificial Neural Network (ANN)

Artificial neural networks are a narrow-sensed abstraction of the human brain, thus the organization of the artificial neural system is very similar to the one of biological neurons. The comprehensive understanding of biological neurons is not complete; however, the basic functionality that contributes to the learning ability of a system is implemented in artificial neural networks. The fundamental element, an artificial neuron, is a model based on known behavior of biological neurons that exhibit most of the characteristics of human brains that we are interested in [Vel98]. This is the most significant difference from conventional computers, which have internal fixed instructions to perform specific functions.

Artificial neural networks can be also described as highly parallel distributed computing models. The fundamental processing units, neurons, are highly connected with strengths, which are dynamically changed during the system's learning process.

The earliest work in neural computing goes back to the 1940's when McCulloch and Pitts [2] introduced the first neural network computing model. Originally in 1943 Warren S. McCulloch, a neuroscientist, Walter Pitts, a logician, published their first paper [2].

In this paper McCulloch and Pitts (MCP) tried to understand how the brain could produce highly complex patterns by using many basic cells that are connected together. These basic brain cells are called neurons, and McCulloch and Pitts gave a highly simplified model of a neuron in their paper. The McCulloch and Pitts model of a neuron, which we will call an MCP neuron for short, has made an important contribution to the development of artificial neural networks -- which model key features of biological neurons.

The original MCP Neurons had limitations. Additional features were added which allowed them to “learn.” The next major development in neural networks was the concept of a perceptron which was introduced by Frank Rosenblatt in 1958. Essentially the perceptron is an MCP neuron where the inputs are first passed through some “preprocessors,” which are called association units. These association units detect the presence of certain specific features in the inputs. In fact, as the name suggests, a perceptron was intended to be a pattern recognition device, and the association units correspond to feature or pattern detectors.

The original MCP Neurons had limitations. Additional features were added which allowed them to “learn.” The next major development in neural networks was the concept of a perceptron which was introduced by Frank Rosenblatt in 1958. Essentially the perceptron is an MCP neuron where the inputs are first passed through some “preprocessors,” which are called association units. These association units detect the presence of certain specific features in the inputs. In fact, as the name suggests, a perceptron was intended to be a pattern recognition device, and the association units correspond to feature or pattern detectors.

The early model of an artificial neuron which was introduced by Warren McCulloch and Walter Pitts is also known as Linear Threshold Gate (LTG) or sometime it is called, the Threshold Logic Unit (TLU). It is a neuron of a set of inputs $I_1, I_2, I_3, \cdots, I_m$ and one output y. The linear threshold gate simply classifies the set of inputs into two different classes. Thus, the output y is a binary form. Such a function can be described mathematically using these equations:

$$Sum = \sum_{i=1}^{N} I_i W_i \qquad \text{Eq. 3-1}$$

and

$$y = f(Sum) \qquad \text{Eq. 3-2}$$

In Equation 3-1, parameters $W_{1,}, W_2, W_3, \cdots, W_m$ are weight values normalized in the range of either (0,1) or (1,-1) and associated with each input line, and the Sum is the

weighted sum, while the function f is a linear step function at threshold T, as shown in Figure 3-1 below.

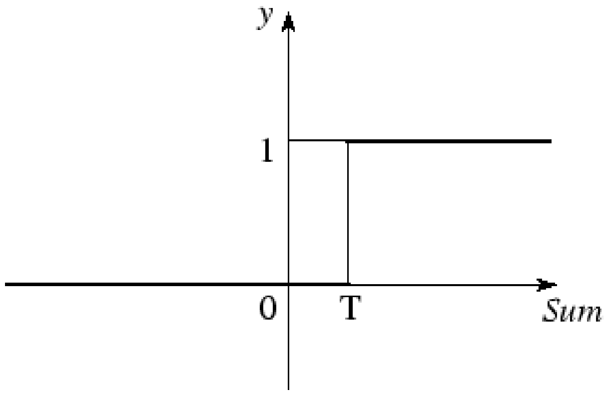


Figure 3-1. Linear Threshold Function.

In Figure 3-1: the value of T, is presentation of a threshold constant. The symbolic representation of the Linear Threshold Gate (LTG) is shown in Figure 3-2

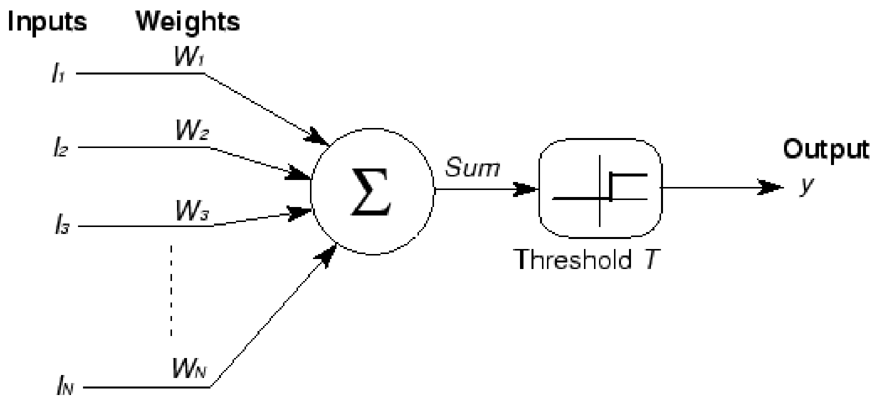


Figure 3-2. Symbolic Illustration of Linear Threshold Gate [3].

The McCulloch-Pitts model of a neuron is simple yet has substantial computing potential. It also has a precise mathematical definition. However, this model is so simplistic that it only generates a binary output and also the weight and threshold values are fixed. The neural computing algorithm has diverse features for various applications [4]. Thus, we need to obtain the neural model with more flexible computational features.

In the 1950's, Rosenblatt's work resulted in a two-layer network, the Perceptron, which was capable of learning certain classifications by adjusting connection weights. Although the perceptron was successful in classifying certain patterns, it had a number of limitations. The perceptron was not able to solve the classic Exclusive Or (XOR) problem (Figure 3-3). Such limitations led to the decline of the field of neural networks. However, the Perceptron had laid foundations for later work in neural computing.

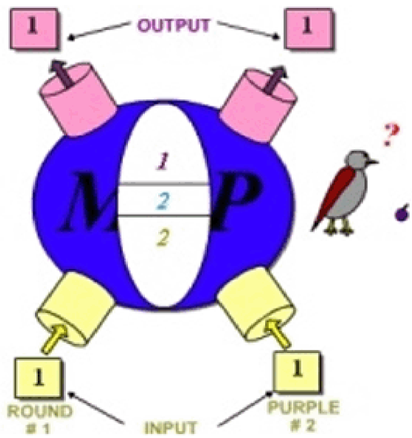


Figure 3-3. Illustration of Exclusive Or (XOR).

Note that: In machine learning, the Perceptron is an algorithm for supervised learning of binary classifiers (functions that can decide whether an input, represented by a vector of numbers, belongs to some specific class or not).

Frank Rosenblatt early 1950 work was consistent of the Perceptron and introduction of a network, which was composed of the units and it was enhanced version of McCulloch-Pitts Threshold Logic Unit (TLU) model. Rosenblatt's model of neuron, a perceptron, was the result of merger between two concepts from the 1940s, McCulloch-Pitts model of an artificial neuron and Hebbian learning rule of adjusting weights [5]. In addition to the variable weight values, the perceptron model added an extra input that represents bias. Thus, the modified form of Equation 3-1 is now as follows:

$$Sum = \sum_{i=1}^{N} I_i W_i + b \qquad \text{Eq. 3-3}$$

In Equation 3-3, the constant b represents the bias value.

In the early 1980's, researchers showed renewed interest in neural networks. Recent work includes Boltzmann machines, Hopfield nets, competitive learning models, multilayer networks, and adaptive resonance theory models.

3.2.1. Artificial Neuron with Continuous Characteristics

Based on the McCulloch-Pitts model described previously, the general form an artificial neuron can be described in two stages shown in Figure 3-4. In the first stage, the linear combination of inputs is calculated. Each value of input array is associated with its weight value, which is normally between 0 and 1. Also, the summation function often takes an extra input value θ with weight value of 1 to represent threshold or *bias* of a neuron. The summation function will be then performed as,

$$x = \sum_{i=1}^{N} A_i W_i + \theta \qquad \text{Eq. 3-4}$$

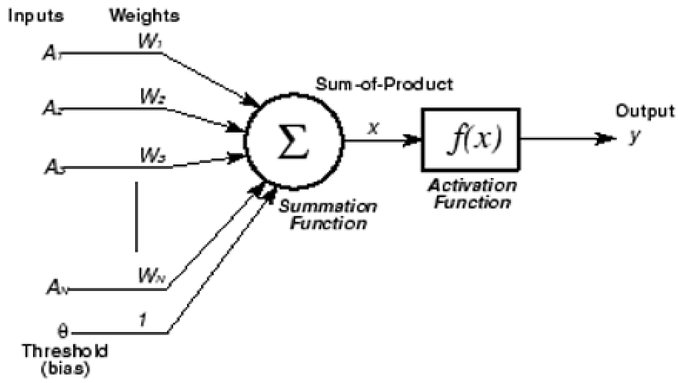


Figure 3-4. More General Neural Model.

The sum-of-product value is then passed into the second stage to perform the activation function which generates the output from the neuron. The activation function "squashes" the amplitude the output in the range of [0,1], or alternately [-1,1] [6]. The behavior of the activation function will describe the characteristics of an artificial neuron model.

The signals generated by actual biological neurons are the action-potential spikes, and the biological neurons are sending the signal in patterns of spikes rather than simple absence or presence of single spike pulse. For example, the signal could be a continuous stream of pulses with various frequencies. With this kind of observation, we should consider a signal to be continuous with bounded range. The linear threshold function should be "softened" [5].

One convenient form of such "semi-linear" function is the logistic sigmoid function, or in short, sigmoid function as shown in Figure 3-5. As the input x tends to large positive value, the output value y approaches to 1. Similarly, the output gets close to 0 as x goes negative. However, the output value is neither close to 0 nor 1 near the threshold point. This function is expressed mathematically as follows:

$$y = \frac{1}{1 + \exp(-x)} \qquad \text{Eq. 3-4}$$

Additionally, the sigmoid function describes the "closeness" to the threshold point by the slope. As x approaches to $-\infty$ or ∞, the slope is zero; the slope increases as x approaches to 0. This characteristic often plays an important role in learning of neural networks.

3.2.2. Single-Layer Network

By connecting multiple neurons, the true computing power of the neural networks comes, though even a single neuron can perform substantial level of computation [6]. The most

common structure of connecting neurons into a network is by layers. The simplest form of layered network is shown in Figure 3-6. The shaded nodes on the left are in the so-called input layer. The input layer neurons are to only pass and distribute the inputs and perform no computation. Thus, the only true layer of neurons is the one on the right. Each of the inputs $x_1, x_2, x_3, \cdots, x_N$ is connected to every artificial neuron in the output layer through the connection weight. Since every value of outputs $y_1, y_2, y_3, \cdots, y_N$ is calculated from the same set of input values, each output is varied based on the connection weights. Although the presented network is fully connected, the true biological neural network may not have all possible connections - the weight value of zero can be represented as "no connection."

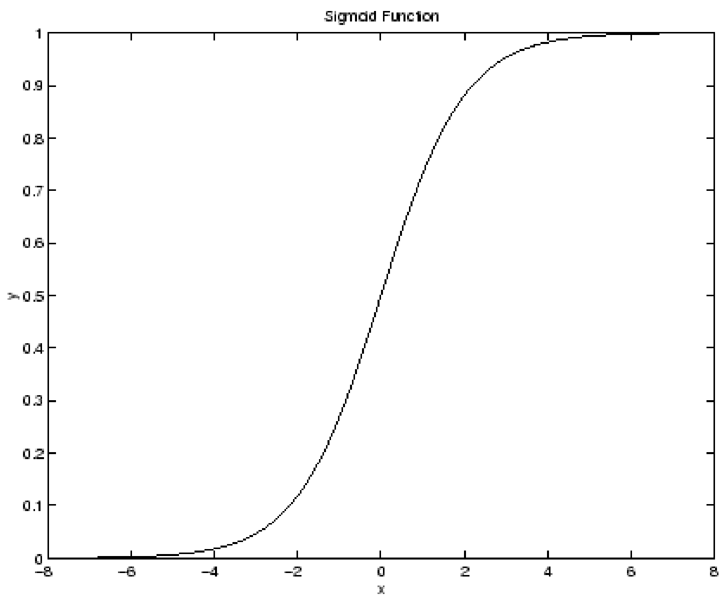


Figure 3-5. Sigmoid Function.

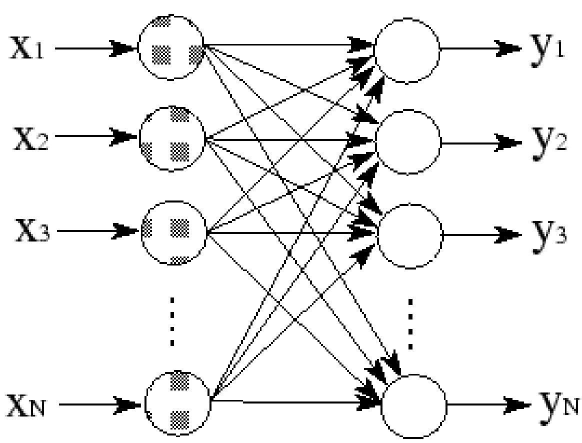


Figure 3-6. Single Layer Neural Network.

3.2.3. Multilayer Network

To achieve higher level of computational capabilities, a more complex structure of neural network is required. Figure 3-7 shows the *multilayer neural network* which distinguishes itself from the single-layer network by having one or more hidden layers. In this multilayer structure, the input nodes pass the information to the units in the first *hidden layer*, then the outputs from the first hidden layer are passed to the next layer, and so on.

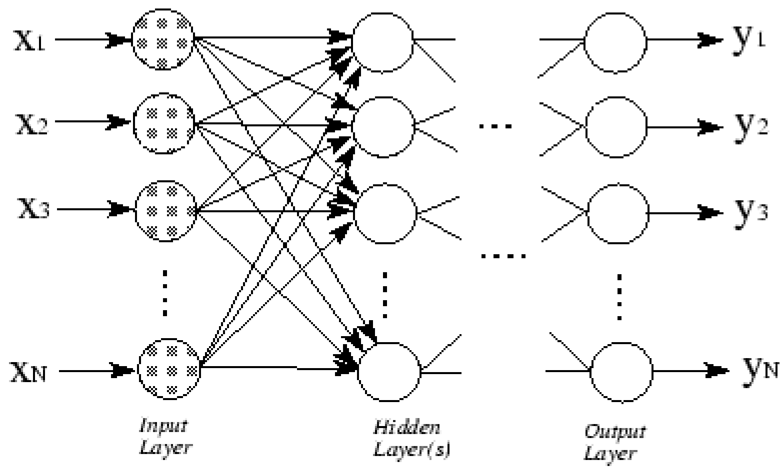


Figure 3-7. Multiple Layer Neural Network.

Multilayer network can be also viewed as cascading of groups of single-layer networks. The level of complexity in computing can be seen by the fact that many single-layer networks are combined into this multilayer network. The designer of an artificial neural network should consider how many hidden layers are required, depending on complexity in desired computation.

3.2.4. Learning Process

Perhaps, the most primary significance of a neural network is the ability to learn the incoming information and to improve the performance of processing information. The term learning refers to many concepts by various viewpoints, and it is difficult to agree on a precise definition of the term. In neural networks, we define learning as the following sequence of events: [Hay 99]

1. Stimulation by an environment in which the network is embedded.
2. Changes in free parameters of the network as the result of stimulation.
3. Responses in a new way to the environment for improved performance.

A Learning algorithm is a prescribed set of well-defined rules for learning of a neural network. There are many types of learning algorithms; the common goal of learning is the adjustment of connection weights.

There are two classes of learning: supervised and unsupervised learning. Supervised learning requires an external source of information in order to adjust the network. On the other hand, in unsupervised learning, there is no external agent that overlooks the process of learning. Instead, the network is adjusted through internal monitoring of performance. In this thesis, we mainly deal with supervised learning since understanding the backpropagation network, which focuses on supervised learning, is our goal here in this section of this chapter.

3.3. BACK-PROPAGATION NEURAL NETWORKS

Back-Propagation Neural Network (BPNN) employs one of the most popular neural network learning algorithms, the Backpropagation (BP) algorithm. It has been used successfully for wide variety of applications, such as speech or voice recognition, image pattern recognition, medical diagnosis, and automatic controls. One of the most striking early applications was NETTalk by T. J. Sejnowski and C. R. Rosenberg in 1986 [8]. The NETTalk was able to learn the rules of phonetics, then the system produced a sound by reading from the sequence of given letters, with a behavior of a child learning to read aloud [9].

Back-propagation made a tremendous step forward from the single-layer perceptron network. With a more sophisticated learning rule, backpropagation networks overcome the limitations that single-layer networks have. Back-propagation is also the most suitable learning method for multilayer networks. Perhaps, the reason why the backpropagation made the major turning point is because the learning rule has a solid mathematical foundation and it is practical [10].

Overall the BPNN, as it is illustrated in Figure 3-8 was developed by Rumelhart et al. [11] as a solution to the problem of training multi-layer perceptrons. The fundamental advances represented by the BPNN were the inclusion of a differentiable transfer function at each node of the network and the use of error back-propagation to modify the internal network weights after each training epoch.

As it is depicted in the Figure 3-8, we see a schematic of a back-propagation neural network. The back-propagation neural networks used in this work all have three layers of neurons, or nodes (input, hidden, and output). Each node in the input and hidden layers is connected to each of the nodes in the next layer (hidden or output). All connections between nodes are directed (i.e., the information flows only one way), and there are no connections between the nodes within a particular layer. Each connection between nodes has a weighting factor associated with it. These weights are modified using the back-propagation algorithm during the training process to produce “learning.”

The BPNN was chosen as a classifier primarily because of its ability to generate complex decision boundaries in the feature space [12]. There is even work suggesting that a BPNN, under appropriate circumstances, can approximate Bayesian posterior probabilities at its outputs [13]. This is significant because a Bayesian classifier provides the best performance possible (i.e., lowest error rate) for a given distribution of the feature data. As with other non-parametric approaches to pattern classification, it is not possible to predict the performance of a BPNN a priori. Furthermore, there are several parameters of the BPNN that must be chosen, including the number of training samples, the number of hidden nodes, and the learning rate.

However since we are interested in a real time processing of input data as fast as they come in and filter these wave data information to a trusted one and stream them to the decision making authority based on Service Level Agreement (SLA), as it was discussed in previous chapters, it is very important we use appropriate posterior. Therefore, suggestion is leaning toward Fuzzy Logic System (FLS) as futuristic approach, for new generation of Artificial Intelligence System, in particular if we are going to place them as a Business Resilience System as it has been discusses by Zohuri and Moghaddam [13].

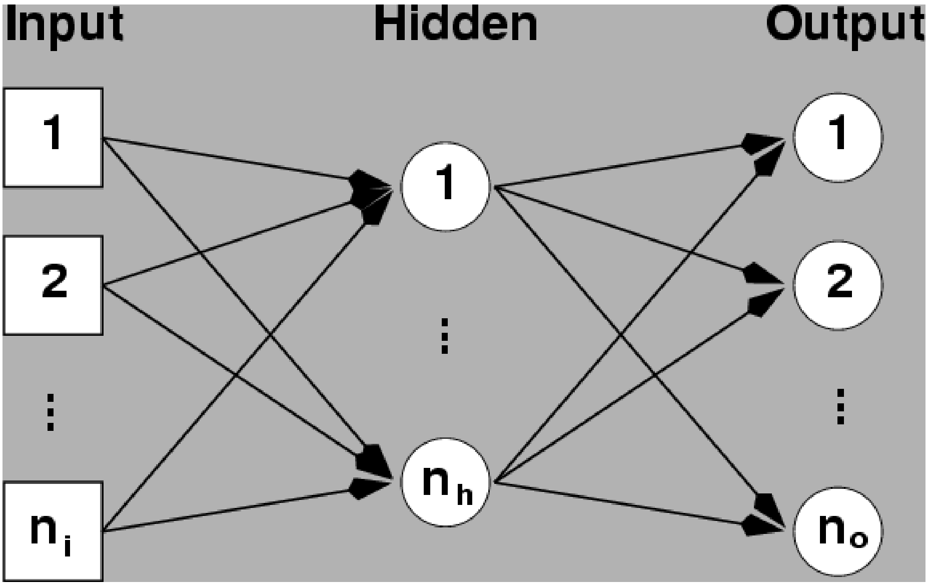


Figure 3-8. A Schematic of a Back-Propagation Neural Network (BPNN).

Based on the work of Baum and Haussler [14], it is possible to place a bound (m) on the number of training samples needed to guarantee a particular level of performance on a set of test samples drawn from the same distribution as the training data. Specifically, if at least m samples are used to train a network with W weights and N nodes such that a fraction equal to $1-\varepsilon/2$ of them are classified correctly, then one can be confident that a fraction $1-\varepsilon$ of future (test) samples from the same distribution will be classified correctly, where

$$m \geq O\left(\frac{W}{\varepsilon}\log\frac{N}{\varepsilon}\right) \qquad \text{Eq. 3-5}$$

As a specific example, to guarantee no more than a 10% error in classifying the test data, the number of training samples should be equal to roughly 10 times the number of weights in the network. For a typical network generated below, this represents a requirement for 5000-10000 training samples. It is simply not tractable to generate that many images. Fortunately, this bound does not preclude the possibility of generating a successful classifier using fewer training samples, as many studies have empirically demonstrated.

The theoretical basis for selecting the number of hidden nodes to use in a single hidden layer network is not well developed. The only general method available to optimize this parameter is to test the network with various numbers of hidden nodes and select the one that performs best.

3.3.1. Linear Separability and the XOR Problem

Consider two-input patterns (X_1, X_2) being classified into two classes as shown in Figure 3-9. Each point with either symbol of x or o represents a pattern with a set of values (X_1, X_2). Each pattern is classified into one of two classes. Notice that these classes can be separated with a single line L. They are known as *linearly separable* patterns. *Linear separability* refers to the fact that classes of patterns with n-dimensional vector $\vec{\boldsymbol{x}} = (x_1, x_2, x_3, \cdots, x_n)$ can be separated with a single decision surface. In the case above, the line L represents the decision surface.

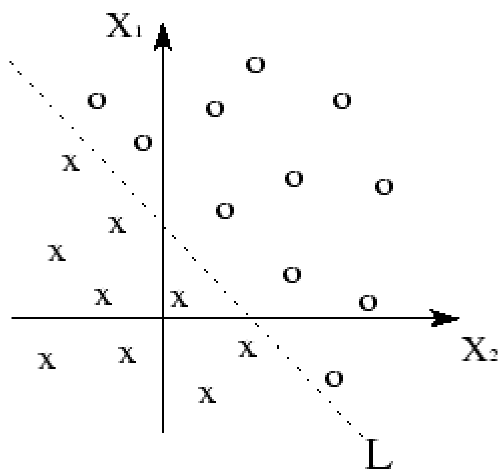


Figure 3-9. Linearly Separable Pattern.

The processing unit of a single-layer perceptron network is able to categorize a set of patterns into two classes as the linear threshold function defines their linear separability. Conversely, the two classes must be linearly separable in order for the perceptron network to function correctly [6]. Indeed, this is the main limitation of a single-layer perceptron network.

The most classic example of linearly inseparable pattern is a logical Exclusive-OR (XOR) function. Shown in Figure 3-10 is the illustration of XOR function that two classes, 0 for black dot and 1 for white dot, cannot be separated with a single line. The solution seems that patterns of (X_1, X_2) can be logically classified with two lines L_1 and L_2 [19].

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

$$Y = X_1 \oplus X_2$$

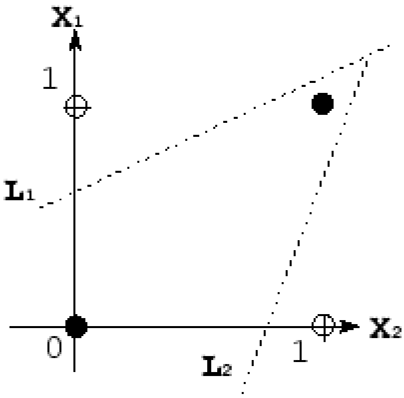


Figure 3-10. Exclusive-OR Function.

3.3.2. Architecture of Backpropagation Networks

Our initial approach to solving linearly inseparable patterns of XOR function is to have multiple stages of perceptron networks. Each stage would set up one decision surface or a line that separate patterns. Based on the classification determined by the previous stage, the current stage can form sub-classifications. Figure 3-11 shows the network with two layers of perceptron units to solve the XOR problem [16]. Node 1 detects the pattern for (1,0), while node 2 detects the pattern for (0,1). Combined, with these first-layer classifications, node 3 is allowed to classify XOR input patterns correctly [20].

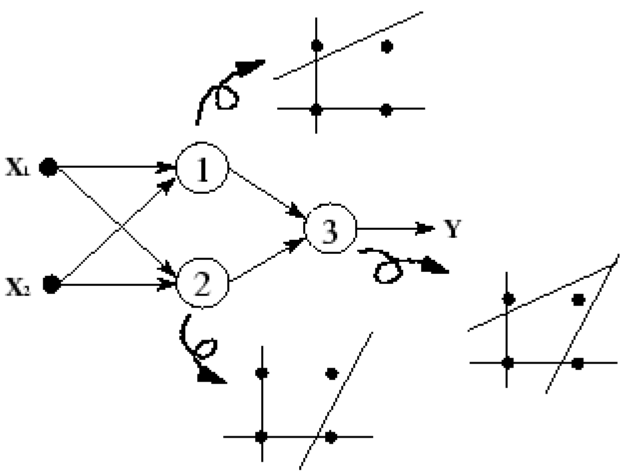


Figure 3-11. Suggested Network for Solving XOR Problem.

Generalizing the XOR case discussed above, the multilayer feed-forward network seems to be the feasible network architecture for backpropagation. However, we still have to take into account how the learning is processed. Unfortunately, with multilayer perceptrons, the nodes in the output layer do not have access to input information in order to adjust connection weights. Because the actual input signals are masked off by the intermediate layers of threshold perceptrons, there is no indication of how close they are to the threshold point. For

this reason, we need to modify a hard-limiting threshold function of the perceptron into a nonlinear function for backpropagation learning.

3.3.3. Back Propagation Processing Unit

The backpropagation processing unit should be in the form modified from a linear perceptron so that the activation function is nonlinear and smoothed out at the threshold point. The suggested form of the activation function is the sigmoid function as mentioned previously. With sigmoid function, we can obtain not only output from the neuron but also information about how close we are to the threshold point using the slope of the sigmoid function. Mathematically, we can derive the slope from the Equation 3-4 as follows:

$$\begin{aligned}\frac{d}{dx} f(x) &= \frac{\exp(-x)}{[1+\exp(-x)]^{-2}} \\ &= \frac{1}{1+\exp(-x)} \frac{\exp(-x)}{1+\exp(-x)} \\ &= \frac{1}{1+\exp(-x)}\left[1-\frac{1}{1+\exp(-1)}\right] \\ &= f(x)[1-f(x)]\end{aligned} \qquad \text{Eq. 3-6}$$

This will be the key information for the weight adjustments in the forthcoming discussions.

3.3.4. Back Propagation Learning Algorithm

The backpropagation algorithm trains a given feed-forward multilayer neural network for a given set of input patterns with known classifications. When each entry of the sample set is presented to the network, the network examines its output response to the sample input pattern. The output response is then compared to the known and desired output and the error value is calculated. Based on the error, the connection weights are adjusted. The backpropagation algorithm is based on *Widrow-Hoff delta learning rule* in which the weight adjustment is done through *mean square error* of the output response to the sample input [21]. The set of these sample patterns are repeatedly presented to the network until the error value is minimized.

Delta Rule

In machine learning, the *delta rule* is a gradient descent learning rule for updating the weights of the inputs to *artificial neurons* in a *single-layer neural network*. It is a special case of the more general backpropagation algorithm. For a neuron j, with activation function $g(x)$, the delta rule for W_{ji}, is given by

Refer to the Figure 3-12 that illustrates the backpropagation multilayer network with M layers. N_j represents the number of neurons in jth layer. Here, the network is presented the p th pattern of training sample set with N_0-dimensional input $X_{p_1}, X_{p_2}, \cdots, X_{p_{N_0}}$ and N_M-dimensional known output response $T_{p_1}, T_{p_2}, \cdots, T_{p_{N_M}}$. The actual response to the input pattern by the network is represented as p $O_{p_1}, O_{p_2}, \cdots, O_{p_{N_M}}$. Let Y_{ji} be the output from the i th neuron in layer j for p th pattern; W_{jik} be the connection weight from kth neuron in layer $(j-1)$ to i th neuron in layer j; and δ_{ji} be the error value associated with the i th neuron in layer j.

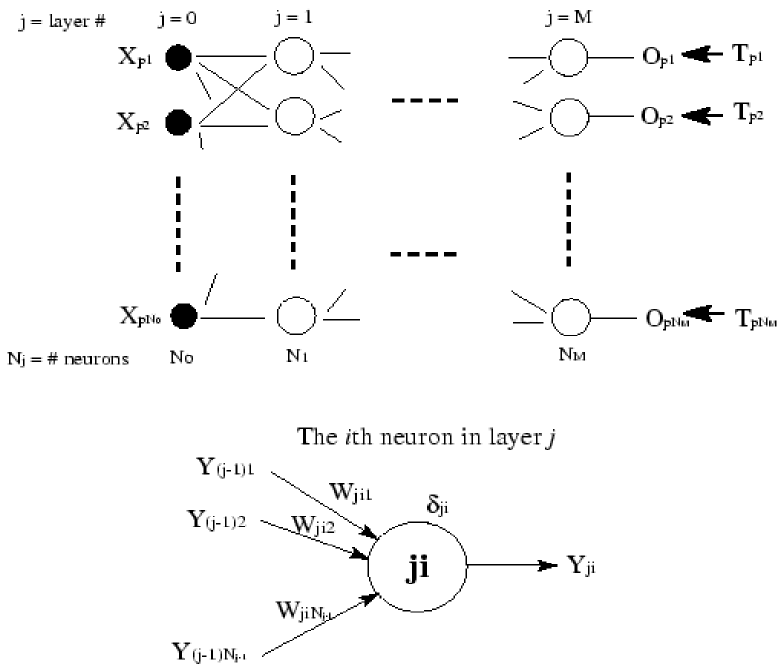


Figure 3-12. Backpropagation Neural Network.

The following is the outline of the backpropagation learning algorithm: [16]

1. Initialize connection weights into small random values.
2. Present the p th sample input vector of pattern $\vec{X}_P = (X_{P_1}, X_{P_2}, \cdots, X_{p_{N_0}})$ and the corresponding output target $\vec{T}_P = (T_{P_1}, T_{P_2}, \cdots, T_{p_{N_M}})$ to the network.
3. Pass the input values to the first layer, layer 1. For every input node i in layer 0, perform:

$$Y_{0i} = X_{p_i} \qquad \text{Eq. 3-7}$$

1. For every neuron i in every layer $j = 1, 2, \cdots, M$, from input to output layer, find the output from the neuron:

$$Y_{ji} = f\left(\sum_{k=1}^{N_{j-1}} Y_{(j-1)k} W_{jik}\right) \qquad \text{Eq. 3-8}$$

where

$$f(x) = \frac{1}{1 + \exp(-x)}$$

2. Obtain output values. For every output node i in layer M, perform:

$$O_{p_i} = Y_{Mi} \qquad \text{Eq. 3-9}$$

3. Calculate error value δ_{ji} for every neuron i in every layer in backward order $j = M, M-1, \cdots, 2, 1$, from output to input layer, followed by weight adjustment. For the output, the error value is:

$$\delta_{Mi} = Y_{Mi}(1 - Y_{Mi})(T_{p_i} - Y_{Mi}) \qquad \text{Eq. 3-10}$$

and for hidden layers:

$$\delta_{Mi} = Y_{Mi}(1 - Y_{Mi}) \sum_{k=1}^{N_{j+1}} \delta_{(j+1)k} W_{(j+1)ki} \qquad \text{Eq. 3-11}$$

The weight adjustment can be done for every connection from neuron k in layer $(i-1)$ to every neuron i in every layer i:

$$W_{jik}^{+} = W_{jik} + \beta \delta_{ji} Y_{ji} \qquad \text{Eq. 3-12}$$

where β represents weight adjustment factor normalized between 0 and 1. The derivation of the equations above will be discussed soon.

The actions in step 2 through 6 will be repeated for every training sample pattern p, and repeated for these sets until the Root Mean Square (RMS) of output errors is minimized.

We now attempt to derive the error and weight adjustment equations shown above. Let's begin with the Root Mean Square (RMS) of the errors in the output layer defined as:

$$E_p = \frac{1}{2}\sum_{j=1}^{N_M}(T_{pj} - O_{pj})^2 \qquad \text{Eq. 3-13}$$

for the p th sample pattern. In generalized delta rule [16, 18, 19], the error value δ_{ji} associated with the i th neuron in layer j is the rate of change in the RMS error E_p respect to the sum-of-product of the neuron as:

$$\delta_{ji} = -\frac{\partial E_p}{\partial net_{ji}} \qquad \text{Eq. 3-14}$$

where net_{ji} represents the sum of product value. With the chain rule, we can obtain the rate of change in RMS error E_p in response to weight change, we can write:

$$\begin{aligned}\frac{\partial E_p}{\partial W_{jik}} &= \frac{\partial E_p}{\partial net_{ji}}\frac{\partial net_{ji}}{\partial W_{jik}} \\ &= -\delta_{ji}\frac{\partial}{\partial W_{jik}}\left[Y_{(j-1)0}W_{(j-1)i0} + \cdots + Y_{(j-1)k}W_{(j-1)ik} + \cdots\right] \\ &= -\delta_{ji}\frac{\partial}{\partial W_{jik}}Y_{(j-1)k}W_{(j-1)ik} \\ &= -\delta_{ji}Y_{(j-1)k}\end{aligned} \qquad \text{Eq. 3-15}$$

we can state that the weight change is proportional to this value [16].

$$\Delta W_{jik} = \beta\delta_{ji}Y_{(j-1)k} \qquad \text{Eq. 3-16}$$

where β is a constant.

Therefore, weight change can be performed as:

$$W_{jik}^{+} = W_{jik} + \Delta W_{jik} \qquad \text{Eq. 3-17}$$

which should match Equation 3-11

Now let us get back to the Equation 3-14, to find an error value associate with the neuron. Again, using the chain rule, we get the following as:

$$\delta_{ji} = -\frac{\partial E_p}{\partial Y_{ji}} \frac{\partial Y_{ji}}{\partial net_{ji}} \quad \text{Eq. 3-18}$$

For output layer, $j = M$ and $Y_{Mi} = O_{pi}$, then we have

$$\begin{aligned} \delta_{Mi} &= -\frac{\partial E_p}{\partial O_{pi}} \frac{\partial Y_{Mi}}{\partial net_{Mi}} \\ &= -\frac{\partial}{\partial O_{pi}} \left\{ \frac{1}{2} \left[(T_{P_1} - O_{pi})^2 + \cdots + (T_{P_i} - O_{pi})^2 + \cdots \right] \right\} \frac{\partial}{\partial net_{Mi}} f(net_{Mi}) \\ &= -\frac{\partial}{\partial O_{pi}} \left[\frac{1}{2} (T_{P_1} - O_{pi})^2 \right] f'(net_{Mi}) \end{aligned} \quad \text{Eq. 3-19}$$

Using Equation 3-6, we get the following relationship as:

$$\begin{aligned} \delta_{Mi} &= (T_{P_1} - O_{pi})[f(net_{Mi})[1 - f(net_{Mi})]] \\ &= (T_{P_1} - O_{pi})(O_{pi}))(1 - f(net_{Mi}) \end{aligned} \quad \text{Eq. 3-20}$$

This should correspond with Equation 3-10. For error values associated with the hidden layer neurons, we cannot use target values. For this reason, the part $\partial E_p / \partial Y_{ji}$ in Equation 3-18 needs to be found using a different approach. We use the chain rule applied to the sum-of-product values of neurons in the front layer {layer $(j+1)$}, so we have.

$$\begin{aligned} \frac{\partial E_p}{\partial Y_{ji}} &= \frac{\partial E_p}{\partial net_{(j+1)1}} \frac{\partial net_{(j+1)1}}{\partial Y_{ji}} + \frac{\partial E_p}{\partial net_{(j+1)2}} \frac{\partial net_{(j+1)2}}{\partial Y_{ji}} + \cdots \\ &= \sum_{a=1}^{N_{j+1}} \left[\frac{\partial E_p}{\partial net_{(j+1)a}} \frac{\partial net_{(j+1)a}}{\partial Y_{ji}} \right] \\ &= \sum_{a=1}^{N_{j+1}} \left[-\delta_{(j+1)a} \frac{\partial}{\partial Y_{ji}} (W_{(j+1)a0} Y_{j0} + \cdots + W_{(j+1)ai} Y_{ji} + \cdots \right] \\ &= \sum_{a=1}^{N_{j+1}} \left[-\delta_{(j+1)a} \frac{\partial}{\partial Y_{ji}} (W_{(j+1)ai} Y_{ji}) \right] \\ &= \sum_{a=1}^{N_{j+1}} \left[-\delta_{(j+1)a} (W_{(j+1)ai} \right] \end{aligned} \quad \text{Eq. 3-21}$$

Finally, combined with $\partial Y_{ji} / \partial net_{ji}$, we obtain the following relation:

$$\delta_{ji} = -\sum_{a=1}^{N_{j+1}} \left[-\delta_{(j+1)a} (W_{(j+1)ai} \right] \frac{\partial Y_{ji}}{\partial net_{ji}}$$
$$= Y_{ji}(1 - Y_{ji}) \sum_{a=1}^{N_{j+1}} \left[-\delta_{(j+1)a} (W_{(j+1)ai} \right] \quad \text{Eq. 3-22}$$

This should concur with Equation 3-11.

3.3.5. Local Minimum Problem

The backpropagation algorithm, as just described, employs gradient descent by following the slope of RMS error value E_p downward along with the change in all the weight values. The weight values are constantly adjusted until the value of E_p is no longer decreasing. Since the RMS error value is very complex function with many parameter values of weights, it is possible that the backpropagation network may converge into a local minima instead of the desired global minimum. This phenomenon of “learning paralysis” can be avoided with several solutions suggested [23]. One is the matter of order in presenting training samples to the learning network. Adding noise to the weights while being updated could be also the solution. Another answer is to utilize momentum, which gradually increases the weight adjustment rate β. All of these solutions are the way to escape from the trap of a local minimum.

3.3.6. Generalization

A trained backpropagation network is able to detect and classify an input pattern that has not been seen during learning. This feature is called generalization, borrowed from the psychology terms. Neural networks are known to be good at classifying noisy input patterns, but not at classifying a pattern that is intermediate between two solid patterns from the training samples. In other words, neural networks are good at interpolation but not extrapolation [16]. Also, there may exist over fitted input data, the unseen input pattern such that it can be classified into one of the trained output response undesirably. Suggested solution includes modification of network architecture and more adequate training samples [20].

3.4. Biological Background

Cells represent a fundamental level of organization in all eukaryotic organisms (i.e., plants and animals). Simply, a eukaryotic cell is a membrane-bounded compartment containing the molecules and sub-compartments necessary to carry out a particular function.

Note the number of sub-compartments and the complexity of their organization even in the simplified representation of a eukaryotic cell in Figure 3-13. In spite of their diversity and complexity, however, all cells contain the same basic components: DNA, RNA, and protein.

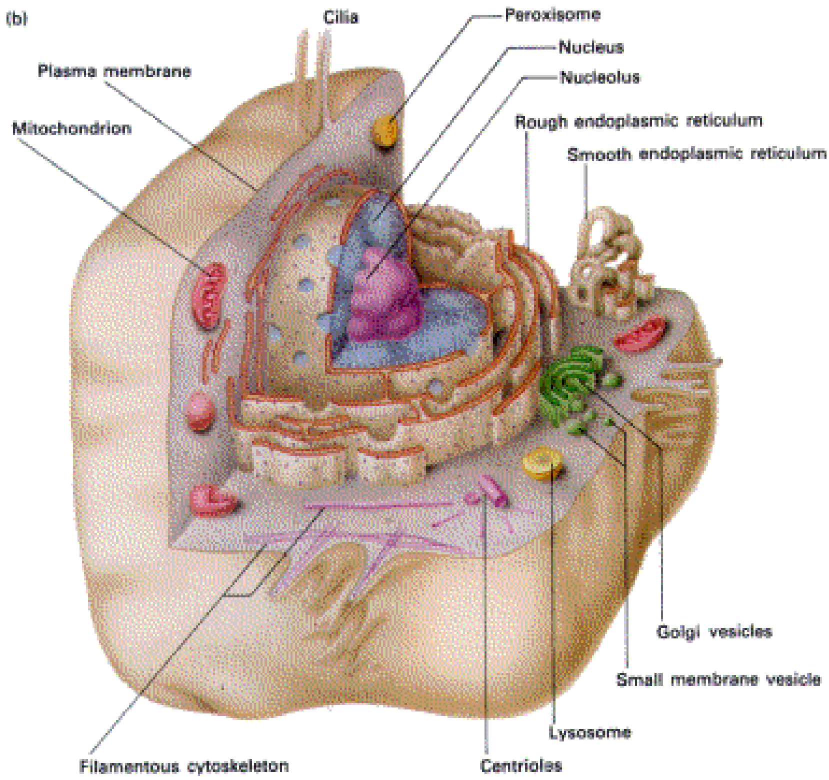


Figure 3-13. A simplified cartoon depicting the basic components of a eukaryotic cell. Reprinted from MOLECULAR CELL BIOLOGY by Lodish et al. ©1986, 1990, 1996 by Scientific American Books, Inc. Used with permission by W.H. Freeman and Company.

All information necessary for the functions of a cell to be carried out is contained in its DNA. A molecule of DNA is simply a long double helix composed of four basic units called nucleotides. The human genome, for instance, is a DNA sequence which is some three *billion* nucleotides long. DNA does not carry out cellular processes on its own, but instead encodes proteins that do. Each protein required by the cell is encoded in a stretch of DNA called a gene. Like DNA, proteins are all made from sequences of basic building blocks. Instead of the four nucleotides of DNA, proteins are made of strings of twenty amino acids. Amino acids are each chemically and structurally unique so that when arranged in long chains and folded properly, they constitute the proteins that provide all of the functionality needed by the cell. Each of the cellular compartments in Figure 1.1 contains many proteins, some unique to that compartment, some not. The combination of proteins in a particular compartment account for the structure, function and localization of that compartment.

For a cell to produce a protein, the corresponding DNA sequence (gene) is first transcribed into a molecule of RNA. RNA, like DNA, consists of four nucleotides arranged in a sequence. Each successive triplet of nucleotides in the RNA sequence codes for a single amino acid in the protein. Through a process known as translation, the RNA sequence is

converted into an amino acid sequence and therefore a protein. The protein-coding portion of a typical mammalian gene is 2000 to 3000 nucleotides long, corresponding to a protein length of 700-1000 amino acids.

This simple overview of protein production does not begin to address the complexity of cellular functions that are carried out by those proteins. Suffice it to say that the entire field of cell biology is devoted to elucidating these details! Two areas of investigation relevant to the work described below are genomics (the study of genomes) and proteomics (the study of proteomes).

Genomics was born out of the ability to sequence the molecules of DNA in cells. It represents a significant advance because it is one of the first areas of biology to which quantitative analysis has been applied. For some time now it has been possible to rigorously compare DNA sequences (e.g., genes) to one another with the intent of placing a number on the degree to which two sequences are similar. The motivation for this analysis is that genes with similar sequence tend to code for proteins with similar structure and function. From this it follows that one can make a prediction about the structure and function of a new protein based on the similarity of its gene sequence to the gene sequence of proteins that have already been characterized. An important milestone in biology and the current focus of much of the work in genomics is obtaining the complete DNA sequence of the human genome, i.e., the Human Genome Project (HGP) presented by Oak Ridge National Laboratory (ORNL). A complete DNA sequence does not correspond to complete knowledge regarding the many proteins it encodes, however. The field of proteomics has evolved to provide this information.

The Human Genome Project (HGP) was an international scientific research project with the goal of determining the sequence of nucleotide base pairs that make up human DNA, and of identifying and mapping all of the genes of the human genome from both a physical and a functional standpoint [21]. It remains the world's largest collaborative biological project [22]. After the idea was picked up in 1984 by the US government when the planning started, the project formally launched in 1990 and was declared complete in 2003. Funding came from the US government through the National Institutes of Health (NIH) as well as numerous other groups from around the world. A parallel project was conducted outside of government by the Celera Corporation, or Celera Genomics, which was formally launched in 1998. Most of the government-sponsored sequencing was performed in twenty universities and research centers in the United States, the United Kingdom, Japan, France, Germany, Canada, and China [23].

The Human Genome Project (HGP) originally aimed to map the nucleotides contained in a human haploid reference genome (more than three billion). The “genome” of any given individual is unique; mapping the “human genome” involved sequencing a small number of individuals and then assembling these together to get a complete sequence for each chromosome. The finished human genome is thus a mosaic, not representing any one individual. See the logo of HGP, in Figure 3-14.

Although a protein starts out as a sequence of amino acids, it is quickly folded into a complex three dimensional structure (See Figure 3-15). From this structure, a protein derives its unique properties. A protein's amino acid sequence can provide some information about its structure and function, but complex studies involving a variety of disciplines including crystallography, nuclear magnetic resonance (NMR) spectroscopy, and biochemistry are required to properly assign structure and function to a protein. These studies are not yet as `high-throughput' as DNA sequencing, but there is a growing realization that additional

information about each protein will be necessary to complement the raw sequence that will be the product of the Human Genome Project.

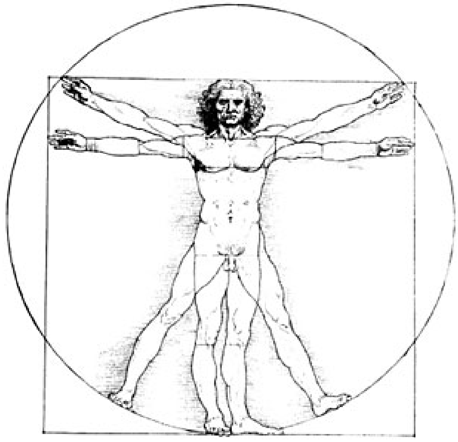
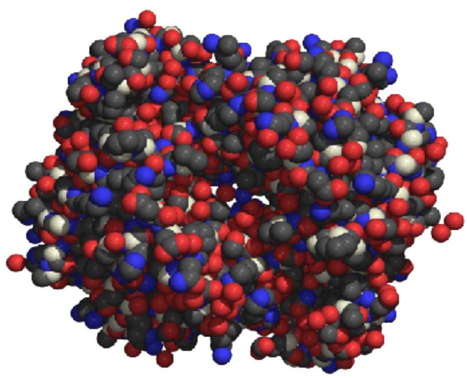
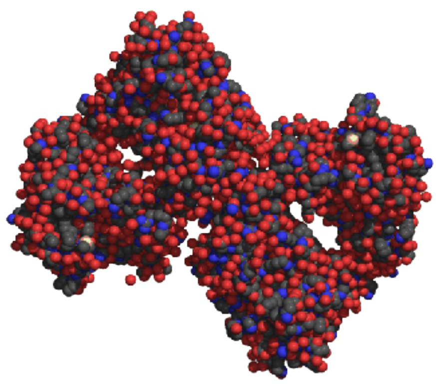


Figure 3-14. Logo HGP; Vitruvian Man, Leonardo da Vinci.



A



B

Figure 3-15. Space-filling models of a single molecule of human hemoglobin (A), and a portion of a mouse antibody like the one used in the experiments described below (B), where each `ball' represents

an atom. Note that the linear sequences of amino acids are folded into complex, compact structures. These images were generated by a combination of the POVChem and POVRay programs and Protein Data Bank (PDB) files obtained from the Research Collaboratory for Structural Bioinformatics.

Even after every gene is sequenced and the structure and function of every protein has been determined, the monumental task of understanding how it all works together inside a cell will still exist. A partial list of the cell functions into which each protein must be placed includes:

1) the general category of metabolism - how does a cell acquire and use energy?
2) locomotion - how do cells change shape and move from one place to another?
3) cell signaling - how does a cell receive input from its environment and how do cells communicate with one another?
4) intracellular transport - how do cells move material from one place to another internally?
5) reproduction - how do cells proliferate?

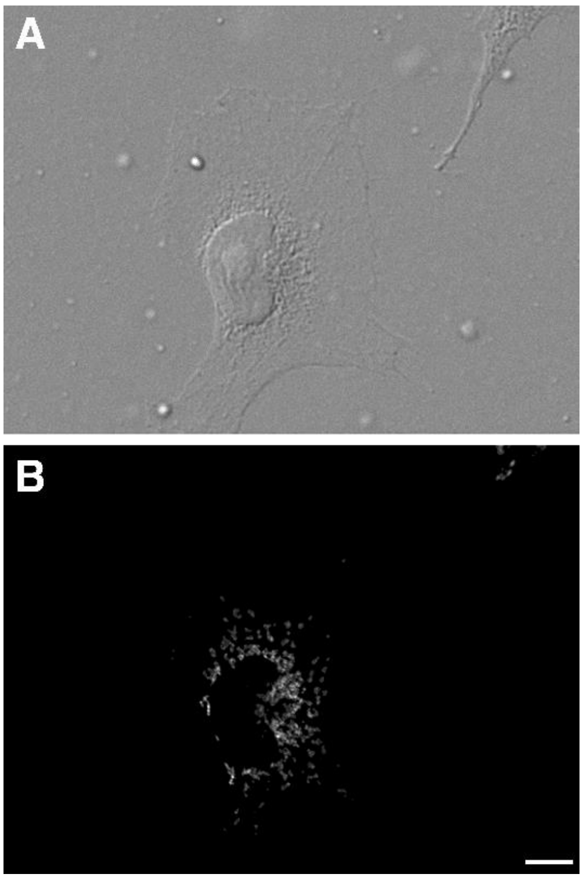


Figure 3-16. Images of the same cell collected using a transmitted light mode (A - Differential Interference Contrast), and a fluorescence mode (B). The fluorescence in B shows the localization of a mitochondrial protein. Scale bar = 10 μm.

One aspect of proteins that is not currently being adequately addressed is their sub-cellular localization (i.e., where within a cell does each protein carry out its function?). To illustrate the concept of sub-cellular localization, Figure 3-16 includes two different images of the same cell. Part A is a transmitted-light image showing the full extent of the cell. Part B is a fluorescence image (See Appendix A of this book, for a discussion of fluorescence) that depicts only the localization of a particular protein, in this case one that is found in the mitochondria. Localization information is important because it provides a context for a protein's structural and functional information. For example, two proteins that possess similar structure and function may in fact be found in distinct compartments within the cell and therefore may be involved in unrelated cellular processes. The work described below is the first of which we are aware that addresses the sub-cellular localization of proteins in a quantitative manner.

3.5. BIOLOGICAL NEURAL NETWORKS

The neural system of the human body consists of three stages: receptors, a neural network, and effectors. The receptors receive the stimuli either internally or from the external world, then pass the information into the neurons in a form of electrical impulses. The neural network then processes the inputs then makes proper decision of outputs. Finally, the effectors translate electrical impulses from the neural network into responses to the outside environment. Figure 3-17 shows the bidirectional communication between stages for feedback

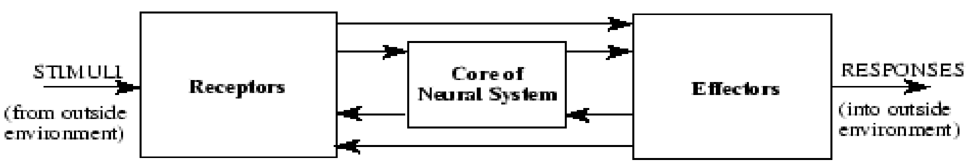


Figure 3-17. Three Stages of Biological Neural System.

The fundamental element of the neural network is called a neuron. As shown in Figure 3-18, a neuron mainly consists of three parts: dendrites, soma, and axon. Dentrites are the tree-like structure that receives the signal from surrounding neurons, where each line is connected to one neuron. Axon is a thin cylinder that transmits the signal from one neuron to others. At the end of axon, the contact to the dendrites is made through a synapse. The inter-neuronal signal at the synapse is usually chemical diffusion but sometimes electrical impulses. A neuron fires an electrical impulse only if certain condition is met [4].

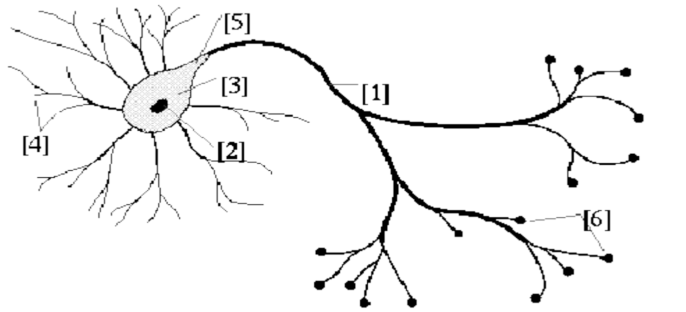


Figure 3-18. A Biological Neuron.

The incoming impulse signal from each synapse to the neuron is either excitatory or inhibitory, which means helping or hindering firing. The condition of causing firing is that the excitatory signal should exceed the inhibitory signal by a certain amount in a short period of time, called the period of latent summation. As we assign a weight to each incoming impulse signal, the excitatory signal has positive weight and the inhibitory signal has negative weight. This way, we can say, "A neuron fires only if the total weight of the synapses that receive impulses in the period of latent summation exceeds the threshold" [24].

3.6. Learning in a Neural Network

Learning is essential to most of these neural network architectures and hence the choice of a learning algorithm is a central issue in network development. What is really meant by saying that a processing element learns? Learning implies that a processing unit is capable of changing its input/output behavior as a result of changes in the environment. Since the activation rule is usually fixed when the network is constructed and since the input/output vector cannot be changed, to change the input/output behavior the weights corresponding to that input vector need to be adjusted. A method is thus needed by which, at least during a training stage, weights can be modified in response to the input/output process. A number of such learning rules are available for neural network models. In a neural network, learning can be supervised, in which the network is provided with the correct answer for the output during training, or unsupervised, in which no external teacher is present.

3.6.1. Pattern Associator

A pattern associator learns associations between input patterns and output patterns. One of the most appealing characteristics of such a network is the fact that it can generate what it learns about one pattern to other similar input patterns. Pattern associators have been widely used in distributed memory modeling.

The pattern associator is one of the more basic two-layer networks. Its architecture consists of two sets of units, the input units and the output units. Each input unit connects to each output unit via weighted connections. Connections are only allowed from input units to

output units. The effect of a unit ui in the input layer on a unit uj in the output layer is determined by the product of the activation ai of ui and the weight of the connection from ui to uj. The activation of a unit uj in the output layer is given by:

$$\sum(W_{ij} \cdot ai) \qquad \text{Eq. 3-23}$$

A pattern associator can be trained to respond with a certain output pattern when presented with an input pattern. The connection weights can be adjusted in order to change the input/output behavior. However, one of the most interesting properties of these models is their ability to self-modify and learn. The learning rule is what specifies how a network changes it weights for a given input/output association. The most commonly used learning rules with pattern associators are the Hebb rule and the Delta rule.

3.6.2. The Hebb Rule

The Hebb rule determines the change in the weight connection from ui to uj by $\Delta W_{ij} = r \cdot ai \cdot aj$, where r is the learning rate and ai, aj represent the activations of ui and uj respectively. Thus, if both ui and uj are activated the weight of the connection from ui to uj should be increased.

Examples can be given of input/output associations which can be learned by a two-layer Hebb rule pattern associator. In fact, it can be proved that if the set of input patterns used in training are mutually orthogonal, the association can be learned by a two-layer pattern associator using Hebbian learning. However, if the set of input patterns are not mutually orthogonal, interference may occur and the network may not be able to learn associations. This limitation of Hebbian learning can be overcome by using the delta rule.

3.6.3. The Delta Rule

Developed by Widrow and Hoff (1960) [25], the delta rule, also called the Least Mean Square (LMS) method, is one of the most commonly used learning rules. For a given input vector, the output vector is compared to the correct answer. If the difference is zero, no learning takes place; otherwise, the weights are adjusted to reduce this difference. The change in weight from ui to uj is given by:

$$\delta W_{ij} = r \cdot ai \cdot ej$$

where r is the learning rate, ai represents the activation of ui and ej is the difference between the expected output and the actual output of uj. If the set of input patterns form a linearly independent set then arbitrary associations can be learned using the delta rule.

It has been shown that for networks with linear activation functions and with no hidden units (hidden units are found in networks with more than two layers), the error squared vs. the weight graph is a paraboloid in n-space. Since the proportionality constant is negative, the graph of such a function is concave upward and has a minimum value. The vertex of this paraboloid represents the point where the error is minimized. The weight vector corresponding to this point is then the ideal weight vector.

This learning rule not only moves the weight vector nearer to the ideal weight vector, it does so in the most efficient way. The delta rule implements a gradient descent by moving the weight vector from the point on the surface of the paraboloid down toward the lowest point, the vertex. Minsky and Papert (1988) [26] raised good questions.

- Is there a simple learning rule that is guaranteed to work for all kinds of problems?
- Does the delta rule work in all cases?

As stated previously, it has been shown that in the case of linear activation functions where the network has no hidden units, the delta rule will always find the best set of weight vectors. On the other hand, that is not the case for hidden units. The error surface is not a paraboloid and so does not have a unique minimum point. There is no such powerful rule as the delta rule for networks with hidden units. There have been a number of theories in response to this problem. These include the generalized delta rule and the unsupervised competitive learning model.

3.6.4. The Generalized Delta Rule

A generalized form of the delta rule, developed by D.E. Rumelhart, G.E. Hinton, and R.J. Williams [27], is needed for networks with hidden layers. They showed that this method works for the class of semi-linear activation functions (non-decreasing and differentiable).

Generalizing the ideas of the delta rule, consider a hierarchical network with an input layer, an output layer and a number of hidden layers. We will consider only the case where there is one hidden layer. The network is presented with input signals which produce output signals that act as input to the middle layer. Output signals from the middle layer in turn act as input to the output layer to produce the final output vector. This vector is compared to the desired output vector. Since both the output and the desired output vectors are known, the delta rule can be used to adjust the weights in the output layer. Can the delta rule be applied to the middle layer? Both the input signal to each unit of the middle layer and the output signal are, known. What is not, known is the error generated from the output of the middle layer since we do not know the desired output. To get this error, back propagate through the middle layer to the units that are responsible for generating that output. The error generated from the middle layer could be, used with the delta rule to adjust the weights.

3.7. FUZZY LOGIC AND NEURAL NETWORKS

Artificial Intelligence based on Artificial Neural Network (ANN) and Fuzzy Logic (FL) works together hand in hand. Artificial neural networks classify and learn rules for fuzzy logic and fuzzy logic infers from unclear neural networks parameter. The latter statement is a network with fast learning capabilities that produces intelligent, crisp output from fuzzy input and/or from fuzzy parameters and avoids time-consuming arithmetic manipulation.

Incorporating fuzzy principles and fuzzy logic in a neural network provides more and more flexibility and robust and resilience system. system; boundaries may be described more generally, not crisply; inputs may be described more vaguely, yet better control may be obtained. The network itself may be fuzzy, not well defined, and able to reconfigure itself for best performance. The power of such machines may be, illustrated with the following "Gedanken" or Thought excrements. Visualize a machine that has learned to analyze scenery, animals, other machines, and other items. A user describes a vague scene in terms of features such as "something as if a tree, about here" and "something like an animal with four legs and a long tail and so tall, there," and so on. Then the machine draws a three-dimensional landscape with a tree and a dog nearby (and perhaps a mountain in the background, with a lake). Then the user may instruct the machine to make corrections to this scene, again in vague language, and the machine immediately projects a three-dimensional scene, very similar to the one the user had in mind. As all that is done, a train with a whistling sound may be crossing the scene (if the parameters are set right) and nearby a frightened bird flies away.

Imagine a machine that is instructed to design a new three-dimensional machine, based on some approximate specifications. Our Gedanken ma- chine designs a model from the vague specifications, simulates the created machine, makes corrections on the model, and, if the corrected one performs as expected, manufactures the first prototype-all in just a few minutes!

Gedanken Experiment or Thought Experiment

Gedanken is a German word for Thought, thus a Gedanken experiment is an experiment *carried out in thought only*.

A thought experiment considers some hypothesis, theory, or principle for the purpose of thinking through its consequences. Given the structure of the experiment, it may or may not be possible to actually perform it, and if it can be performed, there need be no intention of any kind to actually perform the experiment in question.

The common goal of a thought experiment is to explore the potential consequences of the principle in question: "A thought experiment is a device with which one performs an intentional, structured process of intellectual deliberation in order to speculate, within a specifiable problem domain, about potential consequents (or antecedents) for a designated antecedent (or consequent)" (Yeates, 2004, p. 150) [28].

Gedanken experiment, (German: "thought experiment") term used by German-born physicist Albert Einstein to describe his unique approach of using conceptual rather than actual experiments in creating the theory of relativity.

For example, Einstein described how at age 16 he watched himself in his mind's eye as he rode on a light wave and gazed at another light wave moving parallel to his. According to classical physics, Einstein should have seen the second light wave moving at a relative speed of

zero. However, Einstein knew that Scottish physicist James Clerk Maxwell’s electromagnetic equations absolutely require that light always move at 3 × 10^8 meters (186,000 miles) per second in a vacuum. Nothing in the theory allows a light wave to have a speed of zero. Another problem arose as well: if a fixed observer sees light as having a speed of 3 × 10^8 meters per second, whereas an observer moving at the speed of light sees light as having a speed of zero, it would mean that the laws of electromagnetism depend on the observer. But in classical mechanics the same laws apply for all observers, and Einstein saw no reason why the electromagnetic laws should not be equally universal. The constancy of the speed of light and the universality of the laws of physics for all observers are cornerstones of special relativity.

Examples of thought experiments include Schrödinger's Cat, illustrating quantum indeterminacy through the manipulation of a perfectly sealed environment and a tiny bit of radioactive substance, and Maxwell's demon, which attempts to demonstrate the ability of a hypothetical finite being to violate the 2nd law of thermodynamics.

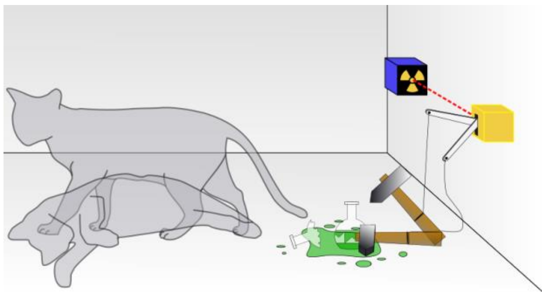


Figure 3-19. Schrödinger's Cat.

Schrödinger's cat (1935) presents a cat that is indeterminately *alive* or *dead*, depending on a random quantum event. It illustrates the counterintuitive implications of Bohr's Copenhagen interpretation when applied to everyday objects.

References

[1] David Poole and Alan Mackworth, “Artificial Intelligence, Foundation of Computational Agents,” Published by Cambridge University Press, 2010.

[2] McCulloch and Pitts, “A logical calculus of the ideas immanent in nervous activity” in the Bulletin of Mathematical Biophysics 5:115-133.

[3] Hassoun, Mohamad H. “Fundamentals of Artificial Neural Networks.” The MIT Press, Cambridge, MA, 1995.

[4] Zurada, Jacek M. “Introduction to Artificial Neural System.” West Publishing Company, St. Paul, MN, 1992.

[5] Bose, N. K. and Liang, P. “Neural Network Fundamentals with Graphs, Algorithms, and Applications.” McGraw-Hill, New York, NY, 1996.

[6] Haykin, Simon. “Neural Networks: A Comprehensive Foundation, second edition.” Prentice-Hall, Upper Saddle River, NJ, 1999.

[7] Lerma Sanchez, Leonardo Octavio. “A Neural Network Approach to a Dimensionality Reduction Problem,” Master's thesis, The University of Texas at El Paso, El Paso, TX, 1991.

[8] Terrence J. Sejnowski and Charles R. Rosenberg, "NETtalk: a parallel network that learns to read aloud" The Johns Hopkins University Electrical Engineering and Computer Science Technical Report JHU/EECS-86/01, 32 pp.
[9] Lerma Sanchez, Leonardo Octavio. "A Neural Network Approach to a Dimensionality Reduction Problem.". Master's thesis, The University of Texas at El Paso, El Paso, TX, 1991.
[10] Dayhoff, Judith E. "Neural Network Architecture: An Introduction." Van Nostrand Reinhold, New York, NY, 1990.
[11] D. Rumelhart, G. Hinton, and Williams R., "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533-536, 1986.
[12] Kurt Hornick, Maxwell Stinchcombe, and Halbert White, "Multilayer feed forward networks are universal approximators," *Neural Networks,* vol. 2, pp. 359-366, 1989.
[13] Bahman Zohuri and Masoud Moghaddam "Business Resilience System (BRS): Driven Through Boolean, Fuzzy Logics and Cloud Computation: Real and Near Real Time Analysis and Decision Making System" 1st ed. 2017, Springer Publishing Company.
[14] E.B. Baum and D. Haussler, "What size net gives valid generalization?," *Neural Computation*, vol. 1, no. 1, pp. 151-160, 1989.
[15] Bach, M. "The Design of the UNIX Operating System." Prentice-Hall, Englewood Cliffs, NJ, 1986.
[16] Beale, R. and Jackson, T. "Neural Computing: An Introduction." Hilger, Philadelphia, PA, 1991.
[17] Velasquez, Guillermo. "A Distributed Approach to a Neural Network Simulation Program." Master's thesis, The University of Texas at El Paso, El Paso, TX, 1998.
[18] Dayhoff, Judith E. "Neural Network Architecture: An Introduction." Van Nostrand Reinhold, New York, NY, 1990.
[19] Gurney, Kevin. "An Introduction to Neural Networks." University of Sheffield Press, London, UK, 1997.
[20] Haykin, Simon. "Neural Networks: A Comprehensive Foundation, second edition." Prentice-Hall, Upper Saddle River, NJ, 1999.
[21] Robert Krulwich (2001-04-17). Cracking the Code of Life (Television Show). PBS.
[22] Economic Impact of the Human Genome Project – Battelle" (PDF format). Retrieved 1 August 2013.
[23] Human Genome Project Completion: Frequently Asked Questions." genome.gov.
[24] Arbib, Michael A. "Brains, Machines, and Mathematics: Second Edition." Springer-Verlag, New York, NY, 1987.
[25] Widrow, B., and Hoff, M. E., Jr., 1960, Adaptive switching circuits, in 1960 IRE WESCON Convention Record, Part 4, New York: IRE, pp. 96–104.
[26] M. L. Minsky and S. A. Papert, *Perceptions: An Introduction to Computational Geometry* expanded edition, The MIT Press, Cambridge, MA, 1988.
[27] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, Learning internal representations by error propagation, in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, Foundations, (D. E. Rumelhart, J. L. McClelland, and PDP Research Group, Eds.), Cambridge, MA: MIT Press, chap. 8.
[28] Yates, J. S. (2004). Doing Social Science Research. London, Sage Publications in association with the Open University Press.

Chapter 4

STRUCTURED AND UNSTRUCTURED DATA PROCESSING

The amount of data that is being created and stored on a global level is almost inconceivable, and it just keeps growing. That means there's even more potential to glean key insights from business information – yet only a small percentage of data is actually analyzed. What does that mean for businesses? How can they make better use of the raw information that flows into their organizations every day?. With respect to the mass categorization that is central to most computer operations, there are two types of relevant data, which affect speed of assimilation as well as information recall: structured data and unstructured data. Smart robots needs both type of data sort and process these data as fast they receive them to level of trusted degree for their processing procedure and set assignment known as service level agreement.

4.1. INTRODUCTION

For the most part, structured data refers to information with a high degree of organization, such that inclusion in a relational database is seamless and readily searchable by simple, straightforward search engine algorithms or other search operations; whereas unstructured data is essentially the opposite. The lack of structure makes compilation a time and energy-consuming task. It would be beneficial to a company across all business strata to find a mechanism of data analysis to reduce the costs unstructured data adds to the organization.

Of course; if it was possible or feasible to instantly transform unstructured data to structured data, then creating intelligence from unstructured data would be easy. However, structured data is akin to machine-language, in that it makes information much easier to deal with using computers; whereas unstructured data, loosely speaking is usually for humans, who do not easily interact with information in strict, database format.

Email is an example of unstructured data; because while the busy inbox of a corporate human resources manager might be arranged by date, time or size; if it were truly fully structured, it would also be arranged by exact subject and content, with no deviation or spread

– which is impractical, because people don't generally speak about precisely one subject even in focused emails.

Spreadsheets, on the other hand, would be considered structured data, which can be quickly scanned for information because it is properly arranged in a relational database system. The problem that unstructured data presents is one of volume; most business interactions are of this kind, requiring a huge investment of resources to sift through and extract the necessary elements, as in a web-based search engine. Because the pool of information is so large, current data mining techniques often miss a substantial amount of the information that's out there, much of which could be game-changing data if efficiently analyzed.

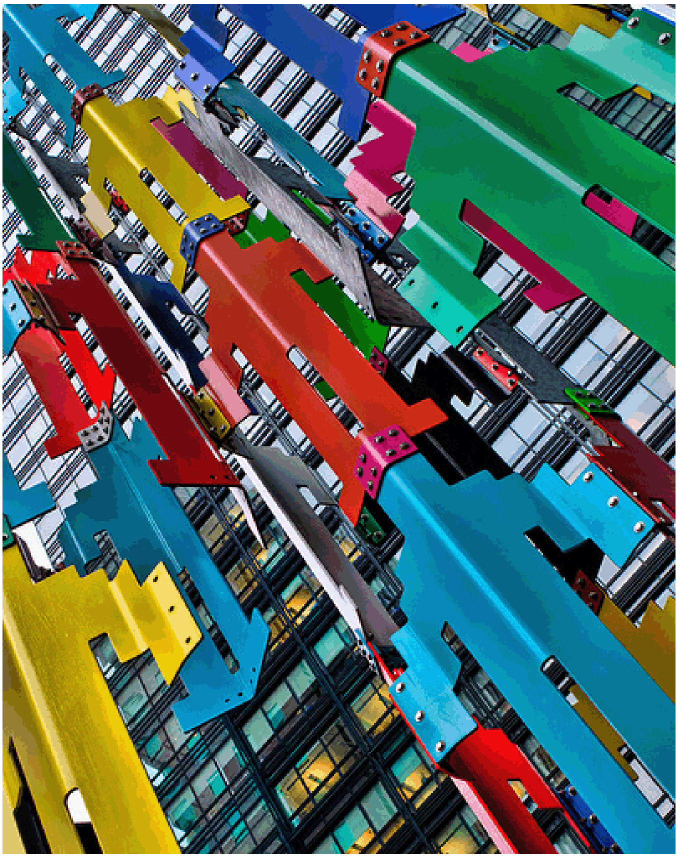


Figure 4-1. Presentation of Unstructured Data Similarity.

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

Harvesting a platform that provides a robust solution for collecting both structured and unstructured data from the Internet. We need to takes a unique approach to "connecting" those unconnected strands of information through the use of metadata. We need harvesting technology that employs multiple threads to mass-harvest scalable quantities of unstructured

data such as web-based central processing, if you will. Harvests should be based on multiple user-developed queries with results (web pages, PDF"s, XLS, PPT, XML, etc.) qualified through customizable filters. This should be developed four scoring algorithms that index the information based on relevancy to further qualify the documents returned, ensuring the user is seeing only super-relevant content. The final user interface displays the qualified results in a searchable database based on customizable facets (URL, file-type, source category, people mentioned, places mentioned, companies mentioned, custom keywords, etc.).

Finding a way to analyze and create intelligence from the wealth of unstructured data available on the Web can be expected to endow an organization with the direct benefit of drastic increases in overall effectiveness and speed of decision making and implementation.

4.2. MASTER DATA MANAGEMENT VERSUS BIG DATA

Initially, Master Data Management (MDM) systems and the content they contain may seem counterintuitive or even diametrically opposed to Big Data systems. Some of the considerable differences between Master Data and Big Data include: [1]

- **Volume:** Comparatively, Master Data sets are much smaller than those for Big Data. One of the pivotal attractions for Big Data is that it encompasses enormous volumes; a person could argue that one of the points of attraction for Master Data is the opposite.
- **Structure:** Master Data tends to contain structured data, while the majority of Big Data is either unstructured or semi-structured.
- **Relationship to the Enterprise:** Typically, MDM systems contain an organization's most trusted data, which tends to be internal, while Big Data platforms quarter enormous amounts of external data from any number of cloud, social media, mobile, and other sources beyond the enterprise's firewall. As indicated by Gartner Group, "MDM is more oriented around internal, enterprise-centric data; in an environment the organization feels it has a chance to effect change, and so formal information governance."

Despite these differences, there are numerous ways in which Master Data Management can enhance Big Data applications, and in which the latter can do so for the former. The basic paradigm for the relationship between these two types of data pertains to the context offered by Big Data and the trust gleaned from Master Data. These virtues can inform one another equally. According to Forrester, MDM can be:

> "…a hub for context in customer experience – sitting between systems of record and systems of engagement to translate, manage and evolve dynamically the full fidelity of customer identity through interactions directly or as viewed through indirect business processes and supporting activities."

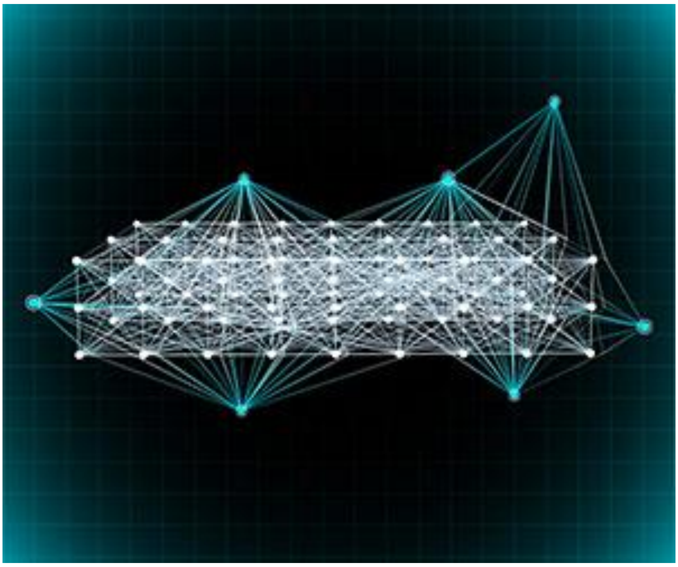


Figure 4-2. Network of Data Volume.

4.2.1. Input: Providing Context to Master Data Management

Organizations can expand their Master Data Management with Big Data by applying the context of data from the external world to their trusted internal data. In this respect, MDM cannot only take advantage of relatively new sources of (Big) Data, but also help provide the proverbial 360 degree, comprehensive view of customers.

Although there are numerous domains for MDM, the customer domain is perhaps most readily enhanced by Big Data. The incorporation of mobile, social, and cloud data can provide numerous points of reference about a customer and his or her experience with an organization’s products that can greatly inform data traditionally stored in MDM. Such data includes customer interactions and relevant transactional data. Thus, Big Data can sufficiently enrich Master Data and facilitate the sort of context that is a critical boon of the former and lead to greater customer understanding. Furthermore, this approach results in Big Data augmenting Master Data to the point where the former is actually aggregated in an MDM hub. Additionally, it is possible to position one’s MDM in the cloud and enable applications to access it as part of Service Oriented Architecture.

Big data is an evolving term that describes any voluminous amount of *structured*, *semistructured* and *unstructured* data that has the potential to be mined for information.

Big data is often characterized by **3Vs**: the extreme ***Volume*** of data, the wide ***Variety*** of data types and the ***Velocity*** at which the data must be processed. Although big data doesn't equate to any specific volume of data, the term is often used to describe terabytes, petabytes and even exabytes of data captured over time.

To break down the 3Vs of big data, we consider, such voluminous data can come from myriad different sources, such as business sales records, the collected results of scientific experiments or *real-time* sensors used in the *internet of things*. Data may be raw or preprocessed using separate software tools before analytics are applied. The internet of things

has been explained in later section of this chapter. However to briefly describe it, we can state that, the Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. See Section 4.5.5 of this chapter.

Data may also exist in a wide variety of file types, including structured data, such as SQL database stores; unstructured data, such as document files; or streaming data from sensors. Further, big data may involve multiple, simultaneous data sources, which may not otherwise be integrated. For example, a *big data analytics* project may attempt to gauge a product's success and future sales by correlating past sales data, return data and online buyer review data for that product.

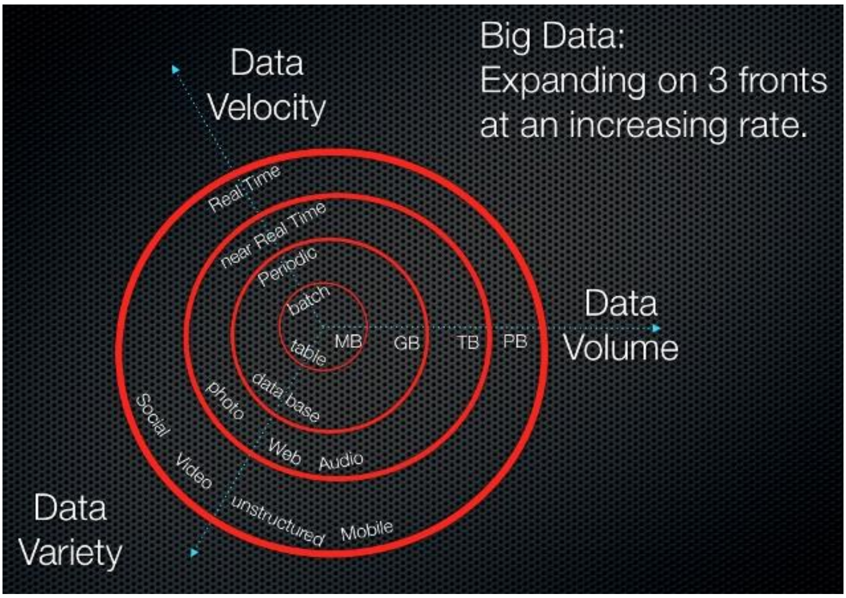


Figure 4-3. Depiction of 3Vs Breakdown.

Finally, velocity refers to the speed at which big data must be analyzed. Every big data analytics project will ingest, correlate and analyze the data sources, and then render an answer or result based on an overarching query. This means human analysts must have a detailed understanding of the available data and possess some sense of what answer they're looking for.

Velocity is also meaningful, as big data analysis expands into fields like machine learning and Artificial Intelligence (AI), where analytical processes mimic perception by finding and using patterns in the collected data.

4.2.2. Big Data Infrastructure Demands

The need for big data velocity imposes unique demands on the underlying compute infrastructure. The computing power required to quickly process huge volumes and varieties of data can overwhelm a single server or server cluster. Organizations must apply adequate

compute power to big data tasks to achieve the desired velocity. This can potentially demand hundreds or thousands of servers that can distribute the work and operate collaboratively.

Achieving such velocity in a cost-effective manner is also a headache. Many enterprise leaders are reticent to invest in an extensive server and storage infrastructure that might only be used occasionally to complete big data tasks. As a result, public cloud computing has emerged as a primary vehicle for hosting big data analytics projects. A public cloud provider can store petabytes of data and scale up thousands of servers just long enough to accomplish the big data project. The business only pays for the storage and compute time actually used, and the cloud instances can be turned off until they're needed again.

To improve service levels even further, some public cloud providers offer big data capabilities, such as highly distributed Hadoop compute instances, data warehouses, databases and other related cloud services. Amazon Web Services Elastic MapReduce is one example of big data services in a public cloud.

4.2.3. The Human Side of Big Data

However these tools only address limited use cases. Many other big data tasks, such as determining the effectiveness of a new drug, can require substantial scientific and computational expertise from the analytical staff. There is currently a shortage of data scientists and other analysts who have experience working with big data in a distributed, open source environment.

Big data can be contrasted with small data, another evolving term that's often used to describe data whose volume and format can be easily used for self-service analytics. A commonly quoted axiom is that “big data is for machines; small data is for people.”

4.2.4. Input: Facilitating Big Data Context to Master Data Management

The challenge with applying Big Data to MDM systems lies in distinguishing relevant unstructured data that relates to Master Data from data which do not. A few options exist for this purpose: vendors have recently implemented NoSQL offerings to attain this end. The distinction in the sheer quantities of data between Master Data and Big Data generally rule out utilizing Hadoop as a means of integrating relevant data, although there are vendors who are working in this vein, as well.

Note that: Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provided massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs. It is as we said an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation. With Hadoop version 2.0's arrival, you will need to explain the benefits of the reigning big data platform to business types and C-level executives.

Basically, Hadoop's second generation offers more to enterprises and simply to understand it, you need to know to fundamental things about it:

1) How it stores files, and
2) How it processes data

What is NoSQL

A NoSQL (originally referring to “non SQL,” “non relational” or “not only SQL”) database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases.

A NoSQL database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases. Such databases have existed since the late 1960s, but did not obtain the “NoSQL” moniker until a surge of popularity in the early twenty-first century, triggered by the needs of Web 2.0 companies such as Facebook, Google, and Amazon.com [3, 4, 5]. NoSQL databases are increasingly used in big data and real-time web applications. NoSQL systems are also sometimes called “Not only SQL” to emphasize that they may support SQL-like query languages.

Motivations for this approach include: simplicity of design, simpler “horizontal” scaling to clusters of machines (which is a problem for relational databases), and finer control over availability. The data structures used by NoSQL databases (e.g., key-value, wide column, graph, or document) are different from those used by default in relational databases, making some operations faster in NoSQL. The particular suitability of a given NoSQL database depends on the problem it must solve. Sometimes the data structures used by NoSQL databases are also viewed as “more flexible” than relational database tables.

Many NoSQL stores compromise consistency (in the sense of the CAP theorem) in favor of availability, partition tolerance, and speed. Barriers to the greater adoption of NoSQL stores include the use of low-level query languages (instead of SQL, for instance the lack of ability to perform ad-hoc JOINs across tables), lack of standardized interfaces, and huge previous investments in existing relational databases [10]. Most NoSQL stores lack true ACID transactions, although a few databases, such as MarkLogic, Aerospike, FairCom c-treeACE, Google Spanner (though technically a NewSQL database), Symas LMDB, and Orient DB have made them central to their designs.

Instead, most NoSQL databases offer a concept of “eventual consistency” in which database changes are propagated to all nodes “eventually” (typically within milliseconds) so queries for data might not return updated data immediately or might result in reading data that is not accurate, a problem known as stale reads. Additionally, some NoSQL systems may exhibit lost writes and other forms of data loss. Fortunately, some NoSQL systems provide concepts such as write-ahead logging to avoid data loss [13]. For distributed transaction processing across multiple databases, data consistency is an even bigger challenge that is difficult for both NoSQL and relational databases. Even current relational databases “do not allow referential integrity constraints to span databases.” There are few systems that maintain both ACID transactions and X/Open XA standards for distributed transaction processing.

A third alternative is the deployment of analytics options (such as those specializing in sentiment data incorporating Natural Language Processing (NLP) (See next section) and other semantic technologies) to first ascertain which data have bearing on germane MDM fields. Aside from recently released MDM solutions that utilize NoSQL methods, it is typically not advantageous to merely add Big Data to an MDM hub without first filtering it. The

aforementioned analytics approach can provide that preliminary point of distinction so that organizations can discern which Big Data can add context to their Master Data.

4.3. Big Data History and Current Considerations

As we said at the beginning of this chapter, the Big Data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it is not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

While the term "big data" is relatively new, the act of gathering and storing large amounts of information for eventual analysis is ages old. The concept gained momentum in the early 2000s when industry analyst Doug Laney articulated the now-mainstream definition of big data as the three Vs as it was described before:

Volume: Organizations collect data from a variety of sources, including business transactions, social media and information from sensor or machine-to-machine data. In the past, storing it would've been a problem – but new technologies (such as Hadoop) have eased the burden.

Velocity: Data streams in at an unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time.

Variety: Data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents, email, video, audio, stock ticker data and financial transactions.

At SAS, we consider two additional dimensions when it comes to big data:

Variability: In addition to the increasing velocities and varieties of data, data flows can be highly inconsistent with periodic peaks. Is something trending in social media? Daily, seasonal and event-triggered peak data loads can be challenging to manage. Even more so with unstructured data.

Complexity: Today's data comes from multiple sources, which makes it difficult to link, match, cleanse and transform data across systems. However, it's necessary to connect and correlate relationships, hierarchies and multiple data linkages or your data can quickly spiral out of control.

4.3.1. Big Data's Big Potential

The amount of data that's being created and stored on a global level is almost inconceivable, and it just keeps growing. That means there's even more potential to glean key insights from business information – yet only a small percentage of data is actually analyzed.

What does that mean for businesses? How can they make better use of the raw information that flows into their organizations every day?

4.3.2. Why Is Big Data Important?

The importance of big data does not revolve around how much data you have, but what you do with it. You can take data from any source and analyze it to find answers that enable:

1) Cost reductions,
2) Time reductions,
3) New product development and optimized offerings, and
4) Smart decision making. When you combine big data with high-powered analytics, you can accomplish business-related tasks such as:
    - Determining root causes of failures, issues and defects in near-real time.
    - Generating coupons at the point of sale based on the customer's buying habits.
    - Recalculating entire risk portfolios in minutes.
    - Detecting fraudulent behavior before it affects your organization.

4.3.3. Who Uses Big Data?

Big data affects organizations across practically every industry. See how each industry can benefit from this onslaught of information.

1. Banking

With large amounts of information streaming in from countless sources, banks are faced with finding new and innovative ways to manage big data. While it's important to understand customers and boost their satisfaction, it's equally important to minimize risk and fraud while maintaining regulatory compliance. Big data brings big insights, but it also requires financial institutions to stay one step ahead of the game with advanced analytics.

2. Government

When government agencies are able to harness and apply analytics to their big data, they gain significant ground when it comes to managing utilities, running agencies, dealing with traffic congestion or preventing crime. But while there are many advantages to big data, governments must also address issues of transparency and privacy.

3. Manufacturing

Armed with insight that big data can provide, manufacturers can boost quality and output while minimizing waste – processes that are key in today's highly competitive market. More and more manufacturers are working in an analytics-based culture, which means they can solve problems faster and make more agile business decisions.

4. Education

Educators armed with data-driven insight can make a significant impact on school systems, students and curriculums. By analyzing big data, they can identify at-risk students, make sure students are making adequate progress, and can implement a better system for evaluation and support of teachers and principals.

5. Health Care

Patient records. Treatment plans. Prescription information. When it comes to health care, everything needs to be done quickly, accurately – and, in some cases, with enough transparency to satisfy stringent industry regulations. When big data is managed effectively, health care providers can uncover hidden insights that improve patient care.

6. Retail

Customer relationship building is critical to the retail industry – and the best way to manage that is to manage big data. Retailers need to know the best way to market to customers, the most effective way to handle transactions, and the most strategic way to bring back lapsed business. Big data remains at the heart of all those things.

For example, we can see big data in action at the United Partial Service (UPS). As a company with many pieces and parts constantly in motion, UPS stores a large amount of data – much of which comes from sensors in its vehicles. That data not only monitors daily performance, but also triggered a major redesign of UPS drivers' route structures. The initiative was called ORION (On-Road Integration Optimization and Navigation), and was arguably the world's largest operations research project. It relied heavily on online map data to reconfigure a driver's pickups and drop-offs in real time. Figure 4-4 is illustration of typical logistic for United Partial Service (UPS).

Figure 4-4. UPS Logistics.

The project led to savings of more than 8.4 million gallons of fuel by cutting 85 million miles off of daily routes. UPS estimates that saving only one daily mile per driver saves the company $30 million, so the overall dollar savings are substantial.

It is important to remember that the primary value from big data comes not from the data in its raw form, but from the processing and analysis of it and the insights, products, and services that emerge from analysis. The sweeping changes in big data technologies and management approaches need to be accompanied by similarly dramatic shifts in how data supports decisions and product/service innovation.

4.3.4. How It Works?

Before discovering how big data can work for your business, you should first understand where it comes from. The sources for big data generally fall into one of three categories:

1. Streaming data

This category includes data that reaches your IT systems from a web of connected devices. You can analyze this data as it arrives and make decisions on what data to keep, what not to keep and what requires further analysis.

2. Social media data

The data on social interactions is an increasingly attractive set of information, particularly for marketing, sales and support functions. It's often in unstructured or semi-structured forms, so it poses a unique challenge when it comes to consumption and analysis.

3. Publicly available sources

Massive amounts of data are available through open data sources like the US government's data.gov, the CIA World Factbook or the European Union Open Data Portal.

After identifying all the potential sources for data, consider the decisions you'll need to make once you begin harnessing information. These include:

1. How to store and manage it

Whereas storage would have been a problem several years ago, there are now low-cost options for storing data if that's the best strategy for your business.

2. How much of it to analyze

Some organizations don't exclude any data from their analyses, which is possible with today's high-performance technologies such as grid computing or in-memory analytics. Another approach is to determine upfront which data is relevant before analyzing it.

3. How to use any insights you uncover

The more knowledge you have, the more confident you'll be in making business decisions. It's smart to have a strategy in place once you have an abundance of information at hand.

The final step in making big data work for your business is to research the technologies that help you make the most of big data and big data analytics. Consider:

- Cheap, abundant storage.
- Faster processors.
- Affordable open source, distributed big data platforms, such as Hadoop.
- Parallel processing, clustering, MPP, virtualization, large grid environments, high connectivity and high throughputs.
- Cloud computing and other flexible resource allocation arrangements

Training the many layers of virtual neurons in the experiment took 16,000 computer processors—the kind of computing infrastructure that Google has developed for its search engine and other services. At least 80 percent of the recent advances in AI can be attributed to the availability of more computer power, reckons Dileep George, cofounder of the machine-learning startup Vicarious.

There's more to it than the sheer size of Google's data centers, though. Deep learning has also benefited from the company's method of splitting computing tasks among many machines so they can be done much more quickly. That's a technology Dean helped develop earlier in his 14-year career at Google. It vastly speeds up the training of deep-learning neural networks as well, enabling Google to run larger networks and feed a lot more data to them.

Already, deep learning has improved voice search on smartphones. Until last year, Google's Android software used a method that misunderstood many words. But in preparation for a new release of Android last July, Dean and his team helped replace part of the speech system with one based on deep learning. Because the multiple layers of neurons allow for more precise training on the many variants of a sound, the system can recognize scraps of sound more reliably, especially in noisy environments such as subway platforms. Since it's likelier to understand what was actually uttered, the result it returns is likelier to be accurate as well. Almost overnight, the number of errors fell by up to 25 percent—results so good that many reviewers now deem Android's voice search smarter than Apple's more famous Siri voice assistant.

For all the advances, not everyone thinks deep learning can move artificial intelligence toward something rivaling human intelligence. Some critics say deep learning and AI in general ignore too much of the brain's biology in favor of brute-force computing.

One such critic is Jeff Hawkins, founder of Palm Computing, whose latest venture, Numenta, is developing a machine-learning system that is biologically inspired but does not use deep learning. Numenta's system can help predict energy consumption patterns and the likelihood that a machine such as a windmill is about to fail. Hawkins, author of On Intelligence, a 2004 book on how the brain works and how it might provide a guide to building intelligent machines, says deep learning fails to account for the concept of time. Brains process streams of sensory data, he says, and human learning depends on our ability to recall sequences of patterns: when you watch a video of a cat doing something funny, it's the motion that matters, not a series of still images like those Google used in its experiment. "Google's attitude is: lots of data makes up for everything," Hawkins says.

But if it does not make up for everything, the computing resources a company like Google throws at these problems can't be dismissed. They're crucial, say deep-learning advocates, because the brain itself is still so much more complex than any of today's neural networks. "You need lots of computational resources to make the ideas work at all.

4.4. REAL TIME DATA PROCESSING AND DATA MINING

Data mining is the process of selecting and exploring data to discover previously unknown patterns and historical trends that can be used to develop models for predicting future outcomes.

Demystifying Data Mining, applies data mining, predictive modeling and real-time analytics in government homeland security, banking, retails, oil and gas operations, etc. Mining large reservoirs of data in oil and gas operations involves committing to key processes and technologies – and embracing new ways of thinking about problem solving. To extract value from vast data stores and change the way decisions are made, many operators have turned to advanced data mining techniques along with real-time analytical and data processing capabilities. This paper explores practical approaches, workflows and techniques that are used in oil and gas operations. It also examines the role of exploratory data analysis; model development and modeling techniques; and approaches to putting models into production.

Data mining is the process of selecting and exploring data to discover previously unknown patterns and historical trends that can be used to develop models for predicting future outcomes. Quantitative techniques uncover patterns and relationships in data that are used to build descriptive and predictive models.

The terms data mining and predictive modeling are often used interchangeably – but they are distinct. Data mining is the process of uncovering patterns in a sample set of data and then developing models that find the same desired pattern across a much larger universe of data. Predictive modeling is the process of applying these models during the course of a business process to predict an outcome.

Analytical technologies have made it possible to understand massive amounts of data to assist in decision making across the enterprise. These same technologies – advanced data mining techniques and real-time analytical and data processing capabilities – can change the way organizations make decisions. And when improved decision making is applied in a structured manner, it can yield significant returns. When these time-tested tools and techniques are adopted by business analysts, they can enable rapid results without requiring the analysts to obtain advanced statistical training.

The scope of activities related to data mining and predictive modeling includes:

- Data preparation to merge multiple data sets, resolve missing values or outliers, and reformat data as needed.
- Exploratory data analysis to discover relationships and anomalies in the data.
- Variable transformation, enrichment and selection to better focus the modeling process.
- Model building using competitive algorithms to search for data combinations that reliably predict the outcome.
- Testing and validation of the champion model to ensure that the model generates output as expected when applied to new data.
- Putting models into production in applications and databases to optimize business processes and improve business decisions.

- Monitoring the model performance to ensure the model is predicting well and does not need to be recalibrated.

4.4.1. Behind the Scenes with Data Mining

Most data aggregation issues are not technical, but rather are related to domain knowledge and data ownership.

The process of data mining includes several key activities. These include data aggregation and preparation, exploratory data analysis, modeling and deployment of models in production environments.

Data preparation is typically 80 percent of the effort of an analytical project. One reason is that many organizations lack a single source of complete, high-quality data required for an analytical exercise. Some choose to wait for the completion of a corporate data warehouse that promises to organize, arrange, standardize and clean the data. Unfortunately, these warehouses seldom address all of the relevant data that may be critical to the analytical problem.

Preparing data for data mining should result in an analytical base table, or data set, that has variables associated with the problem being modeled. This data is prepared very differently than a warehouse for historical reporting because it is gathered from many more databases and stored in very different subsets that are tailored for the analysis at hand.

Aggregation and preparation of data for an analytical project must be conducted by members of the team who have the relevant skill set. Most data aggregation issues are not technical, but rather are related to domain knowledge and data ownership. Once the business logic is provided by the domain experts, the process of accessing, joining and reformatting data can be performed by technical staff.

Data standardization and data quality also require collaboration between domain and technical staff. For instance, standardizing a supplier name across multiple systems is not likely to affect the result of an analytical exercise. However, decisions about how to fill missing range values in sensor tag information or how to collapse time measurement intervals should be made by the domain expert advising the analyst.

Once the workflow for aggregating and cleansing the relevant data is determined, the processes and rules related to conducting these tasks can be automated. Automation of data preparation can be done in real time when the data is created, or at intervals that are appropriate to the data’s specific, time-sensitive requirements.

4.4.2. Modeling Techniques

Best practices for data mining involve utilizing multiple competing analytical techniques to determine which technique will produce the best model and therefore the best prediction. Some of these modeling techniques include decision trees, neural networks, least-angle regressions, logistic regressions, memory-based reasoning and rule induction.

A software solution with multiple modeling techniques allows an analyst to quickly and easily apply a particular modeling technique to a data set and interactively work with the

parameters to try different configurations. This capability allows modelers to test many different approaches while relying on the software to pick the most accurate model, based on a set of user-selected statistical diagnostic tests. It is also important to be able to build an ensemble model by combining techniques if the combination of two models is more effective than a single model. Advanced data mining software that provides these capabilities allows an analyst to spend time developing true insights as opposed to programming analytical models.

Figure 4-5 is illustration of analytical methods that may apply to oil field as an example.

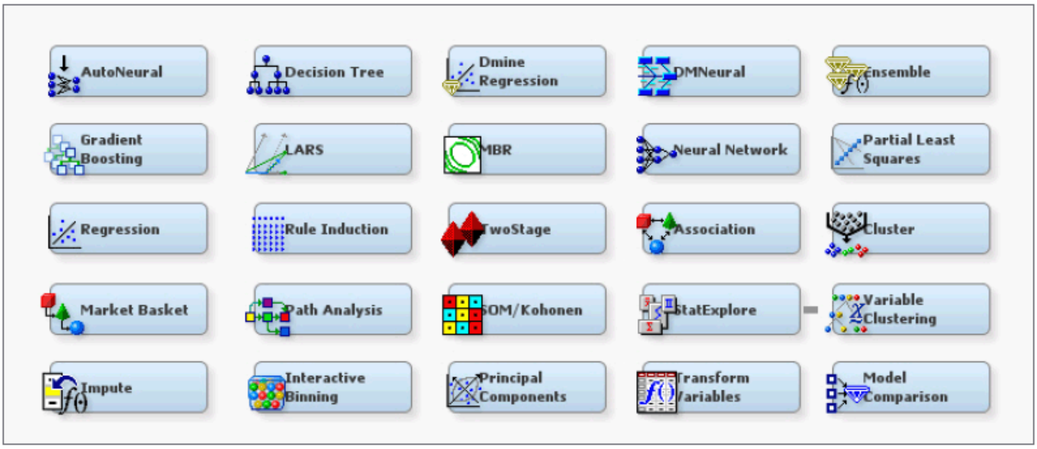


Figure 4-5. Examples of Analytical Methods that Might be Applied to Oilfield Data.

Data mining is most effective when deployed as part of an integrated information delivery strategy that is supported by strong business domain specialists, Information Technology (IT) and skilled analysts.

4.4.3. Deploying Models in a Scalable Environment

Significant business value is lost without the ability to put analytical models into production at an enterprise scale. But putting models into production involves key enabling technologies that must be flexible and robust enough to support the various technical requirements dictated by the business problem. The technical solution must support data access regardless of size and source, and must provide flexible scalability regardless of computational complexity or the time window established to return a computed value.

Data mining is most effective when deployed as part of an integrated information delivery strategy that is supported by strong business domain specialists, IT and skilled analysts. Determining who leads for each data mining exercise requires consideration of the target audience, the timing of the results, and the anticipated action as a result of the analytical insight. This multidisciplinary approach ensures that the project can have technical success and also generate business value.

The technical infrastructure selected for both modeling and production must be able to support modeling at scale. When modeling large data sets, it is important to be able to process large data sets with computationally intensive tasks and to visualize large data sets that support exploratory analysis. In the case of very large data sets or complex computational

processes, it may be necessary to use high-performance computing to return value within a tight time frame.

Just as business outcomes shift with the economy, data constantly changes. As a result, companies should establish models that natively adapt to changes in the data without significant human intervention. It is possible for key variables in a multivariate predictive model to shift in a statistically significant way not anticipated by the model. The ability to monitor and alert key stakeholders to ongoing model performance is the final step in deploying advanced analytical models to scale.

In summary, Data mining technologies can be utilized to exploit vast amounts of data to yield significant results in short time frames. SAS has helped many organizations apply advanced analytics to achieve significant benefits in the digital oilfield. This journey from data exploration to optimized decisions is achievable and can be deployed at scale in oilfield operations.

4.5. Improving Big Data Analytics with Machine Learning-as-a-Service

Machine Learning-as-a-Service (MLaaS) exists as the nexus point for some of the most promising technologies and applications of Big Data analytics. The availability of sophisticated Machine Learning algorithms on demand via the Cloud has significant ramifications for [1]:

- **Big Data:** In many ways, the predictive capabilities of Machine Learning are the only means by which the enterprise can make use of all its data. MLaaS can enable organizations to combine structured data with unstructured, external data to automate analytics processes that would otherwise take too long to parse with Big Data sets.
- **Cognitive Computing:** As one of the central components in Cognitive Computing, Machine Learning helps organizations take a more cognitive approach to their Data Management by producing applications with cognitive capabilities.
- **Data Science:** Machine Learning-as-a-Service has the propensity to both reduce the need for scarcely found Data Scientists and substantially assist them in basing models and business objectives on an organization's data. In either case it eliminates the need to create each individual algorithm.

Figure 4-6. The Smart Robot.

- **Application Development:** The capability of MLaaS to accelerate the Data Science process ensures that developers can create better applications more expeditiously to derive near real-time action from analytics. Figure 4-6 is illustration of Smart Robot as learning machine.
- **Data Modeling:** Once initial algorithms are construed, Machine Learning automates the Data Modeling process by producing models on both present and future data to expedite what otherwise would be a time consuming affair.
- **Cloud Computing:** The speed and convenience of MLaaS are accessed through the Cloud, which reinforces this medium as the architecture of choice for Big Data.

The cumulative effect is that Machine Learning-as-a-Service expands on the possibilities of Big Data analytics while making them more accessible than ever before. James Kobielus of IBM recently indicated that:

> "These value points derive from Machine Learning's core function: enabling analytics algorithms to learn from fresh feeds of data without constant human intervention and without explicit programming."

Machine learning is a type of Artificial Intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data.

The process of machine learning is similar to that of data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension -- as is the case in data mining applications -- machine learning uses that data to detect patterns in data and adjust program actions accordingly. Machine learning algorithms are often categorized as being supervised or unsupervised. Supervised algorithms can apply what has been learned in the past to new data. Unsupervised algorithms can draw inferences from datasets.

For example, Facebook's News Feed uses machine learning to personalize each member's feed. If a member frequently stops scrolling in order to read or "like" a particular friend's posts, the News Feed will start to show more of that friend's activity earlier in the feed. Behind the scenes, the software is simply using statistical analysis and predictive analytics to identify patterns in the user's data and use to patterns to populate the News Feed. Should the member no longer stop to read, like or comment on the friend's posts, that new data will be included in the data set and the News Feed will adjust accordingly.

4.5.1. Application Development

Some of the most eminent Cloud service providers (Amazon, Google, Microsoft, IBM) are offering MLaaS either independently or as part of other platforms. Twitter recently underscored the importance of Machine Learning by acquiring Whetlab, a Machine Learning startup. Perhaps the most immediate of the aforementioned ramifications of MLaaS is the fact that it enables developers to readily incorporate Machine Learning into their applications. Most providers deliver MLaaS as an offering used expressly in conjunction with their Clouds, which underpins the need to facilitate Big Data applications off premises. More importantly,

the ability of developers to utilize Machine Learning algorithms in their applications reduces the reliance on Data Scientists and the complexity associated with creating these algorithms. Applications involving Machine Learning include any assortment of uses from fraud detection and recommender engines to pattern mining, clustering, and other aspects of e-commerce that hinge on Big Data analytics. MLaaS enables developers to take on more responsibility for Data Modeling and Data Science.

4.5.2. Data Science

In addition to reducing the need for Data Scientists, Machine Learning-as-a-Service (MLaaS) can significantly assist these professionals by increasing the complexity of the underlying analytics algorithms that empower the business. One of the key repercussions of MLaaS is that the increased availability of Machine Learning makes it and Big Data less individual concerns, and more integral to overall processes in Data Management. Without having to create each and every algorithm for each and every application, Data Scientists can tackle more profound aspects of *Machine Learning* such as *Deep Learning* and neural networks. In this way, MLaaS becomes as valuable a tool for Data Scientists as it is for organizations without Data Scientists, since it enables the former to expand on the capabilities of this discipline within Data Management.

4.5.3. Cloud Vitality

All of the typical benefits of utilizing the Cloud apply to MLaaS: reduced cost, less infrastructure, increased time to value. Additionally, MLaaS offers the sort of boons that other specialized analytics services such as Graph Analytics-as-a-Service (See next section) provide. Those include a conservation of resources dedicated to hiring specialized personnel and the ability to simply outsource difficult analytics work and its analysis. Certain MLaaS vendors, for instance, will not only construct predictive models and algorithms, but also provide analysis of an organization's data as part of their services. Although each provider's process and capabilities differ, most base charges on the amount of data and the length of time service is used—some have specific prices for different Machine Learning functions. Virtually all of them involve APIs to make data machine readable and provide a framework for basic algorithms and models that organizations can tailor to varying degrees. An assortment of visualization tools and programming languages is supported, while there is a minimal reliance on code.

4.5.4. The Point of Big Data: The Internet of Things

Machine Learning and MLaaS are projected to play an integral role in the facilitation of the Internet of Things (IoT). The IoT is arguably the ultimate expression of Big Data. It enables perpetual connectivity and constant streaming of data from any variety of gadgets, from the industrial to the personal. It would be virtually impossible for a team of Data Scientists to continually refine the algorithms and models required for real-time and

predictive analytics of the immense quantities of data involved in the IoT. The problem with the IoT that Machine Learning ameliorates is the need to not only account for potentially billions of sensors and their constant streaming, but to analyze them in way that produces timely action. Virtually the only way to do so is to build future models and algorithms from historic and real-time data, which is what Machine Learning does. MLaaS expedites that process.

A thing, in the Internet of Things, can be a person with a heart monitor implant, a farm animals with a biochip transponder, and automobile that has built-in sensors to alert the driver when the tire pressure is low--or any other natural or manmade object that can be assigned an IP address and provided with the ability to transfer data over a network.

4.5.5. The Internet of Things

The Internet of Things (IoT) is slated to transform the nature of transactional data from a rigid, performance optimized process, to a dynamic, on-the-fly zone capable of handling the pace and variety of Big Data. There are several reasons to believe Apache Spark will play a formidable role in this transformation:

- In-Memory analytics: Spark utilizes an in-memory analytics approach that is typically faster than most disk based methods—so much so that it has been reported as significantly outperforming Hadoop.
- Machine Learning iterations: This Big Data parallel-processing framework is primed for Machine Learning algorithms that utilize its swiftness for the sort of rapid iterations required to draw conclusions from and make improvements with immense data sets.
- Hadoop compatible: Although some view it as competitive with Hadoop, Spark was actually designed to run on HDFS (but it does not have to). In doing so it can add substantial value to one of the most established and widely used Big Data platforms.

Figure 4-7. Internet of Things.

With vendor support for Spark including MapR, Cloudera, IBM, Data Stax, Intel, Hortonworks, and other Big Data platforms, its effect on Big Data, the IoT, and transactional data is arguably burgeoning throughout the data sphere.

As IoT becomes more widespread it will attract more potential for cyber attacks and fraud. The vast quantity of data that will flow between the connected vehicle, connected home and the insurance company is vulnerable to interception. The new IoT products are also likely to lead to new types of application and claims fraud. Insurers will need to invest more heavily in data security and fraud protection.

While realizing the full potential of IoT for insurance will not be without its challenges, its early exploitation is already producing positive results. IoT undoubtedly makes losses easier to predict and prevent. Smart home devices, wearables and the imminent arrival of the driverless car will usher in a shift toward a new type of customer relationship where insurance will become less reactive and more preventative. The winners will be organizations that overcome today's obstacles to embrace change and capitalize on uncertainty.

The Internet of Things (IoT) is a collection of network-connected physical objects and machines. They have embedded identifications, sensors, and software that can provide an understanding of where they are, what they're doing, and what's going on around them. These devices can communicate with each other and share their data via a network or a cloud-based platform. Examples we hear about most often include a power company's "smart grid" with sensors collectively managing the flow of and demand for electricity, and an individual's "smart home" with climate control, lighting, and security adjusted automatically and remotely.

"The IoT does three things for retailers," said Lori Schafer, Executive Adviser for Retail and CPG at SAS, speaking recently at SAS® Global Forum. "It helps you sense who the customers are and what they're doing; it helps you better understand customer behavior and buying practices; and it allows you to then act upon those insights to create a better overall experience for the customer."

We are moving from a more connected world to a more informed world. And if you're a retailer that isn't informed, you're going to be at a big disadvantage.

4.5.5.1. Additional Benefits

Additionally, Spark has attracted so much attention in the Big Data space because of its benefits:

- Reduced infrastructure: Part of the utility of the high performance speeds of this platform is that they are generated while requiring fewer nodes and clusters than those of more traditional Big Data platforms.
- Open Source aspect: The open source nature of the solution yields up front cost advantages.
- Application building: The diversity of code options greatly assists with application building and decreasing time to value.
- Accelerated performance: Spark's tremendous speed not only facilitates real time analytics, but enables users to incorporate many more data sources than they otherwise would be able to.
- Data Science and developer friendly nature: Although Spark was written in Scala, it contains APIs for popular programming languages such as SQL, Java, and Python. The assortment of code languages is exploited by the platform's underlying engine,

which can determine how to complete jobs on a cluster, thereby ‘hiding’ some of the complexity associated with writing code.

4.5.5.2. Potential Drawbacks

Part of the reason that Spark is unlikely to supplant Hadoop and its ecosystem for the present includes the fact that it:

- Lacks autonomy: Although additional components of the Spark platform include a Semantic Graph analytics engine, a Machine Learning library, and tools for data streaming, its autonomy is circumscribed by a lack of a resource manager or file system, which can simply increase its dependency on Hadoop.
- Is still immature: Having influenced the data landscape over the past five years, Spark still remains a technology that is in the process of maturing.
- Lacks use cases: Enterprise-level deployments of Spark that have demonstrated consistency and performance strengths over time are still in the process of materializing.
- Is not primed for operations: Primed for analytics, Spark is less effective in operations due to rigidity in its core abstraction that produces difficulty in making a single change in a data set, and which frequently requires copying the entire set to do so.

4.5.5.3. Internet on Things Transactions

Current real-time Big Data applications that incorporate elements of both analytics and transactions include standards such as fraud detection, recommender engines, or any variety of Industrial Internet applications pertaining to equipment asset management and preventative maintenance. These applications hint at the value that Spark can provide and the way that the IoT will affect transactions, because they offer a starting point for how the former can already be used.

What the Internet of Things is helping to herald is sophistication in the nature of transactions in which any assortment of analytics (real-time, predictive, and even historic) is required to not only provide insight, but also actually trigger action that expands upon the possibilities of most current transactions. For instance, while most real-time fraud detection systems simply provide approval or denial based on a finite amount of factors, the shift in transactions the IoT is causing can greatly increase the number of those factors. For the consumer, that might mean options for a smart vehicle to obtain gas or for a smart home to reduce power consumption. Regardless of the individual circumstance, the rapidly iterative nature of Spark’s extreme expedience to facilitate Machine Learning algorithms will be well served.

4.5.5.4. Internet on Things Applications Retailers

With IoT, we can now understand the context (the time and the place of the customer) to identify when we are certain the customer needs help or an incentive to purchase, and we can respond proactively.

Key applications of IoT for retailers include supply chain, connected consumer and smart-store applications. In particular, let's look at five areas where retailers are taking advantage of IoT:

1) **Predictive equipment maintenance** is used for managing energy, predicting equipment failure or detecting other issues. For example, every grocery store has a lot of complex equipment – most people recognize refrigeration units. When these units are instrumented with sensors, we can predict maintenance issues that might affect power consumption for savings or monitor temperature fluctuations to ensure food safety.
2) Moving merchandise more efficiently is one of the goals **of smart transportation** applications in retail, and IoT can come into play with the maintenance of transport, tracking and route optimization. We know many retailers have been using GPS to track and route trucks in the last couple of years. With IoT, we are able to understand to a much higher degree of accuracy how close a pallet of merchandise is to a given store.
3) When it comes to **demand-aware warehouse** fulfillment, we're talking about warehouse automation and robotics driven by online and in-store shopping demand. IoT allows us to monitor sales opportunities in real time and track missed in-store sales. It is important to remember that Radio-Frequency IDentification (RFID) is a well-tested part of IoT that can be used for inventory management and more accurate service-level optimization. Currently, a typical distribution center or warehouse is organized by aisles and shelves based on a fixed schematic. The warehouse of the future will be open space where automated pallets self-organize based on real-time demand.
4) Increasingly, the **connected consumer** is having an impact on brick-and-mortar locations. Retailers understand that customers are able to check in-store pricing and local inventory levels from their mobile devices. Imagine if we could make a customized best-price offer or provide location-based services right in the store. What if we could target our high-value, loyal customers with concierge services? In the past, it was accepted as the norm that we would send mass promotions to customers with the expectation that some acceptable percentage might be interested in that promotion. With IoT, we can now understand the context (the time and the place of the customer) to identify when we are certain the customer needs help or an incentive to purchase, and we can respond proactively.
5) In a **smart store**, mall traffic can be analyzed across several retailers so we understand the entire shopping journey. In the past, we had to run expensive survey projects to understand if store associates were being responsive to customer service needs and then enact elaborate staff training programs. Now, within smart stores, we will be able to use video or Wi-Fi foot-traffic monitoring to see if customers dwell over a product area. Then, in real time, direct an associate to help that customer or analyze that information later to adjust store layouts for more efficient customer visits. In addition, by monitoring store traffic and customer demand in real time, we can customize the current in-store shopping experience. That gives us the opportunity to implement rich digital marketing inside the store or announce events to customers via their mobile devices.

With the rapid growth of online shopping, retailers are very keen to bring the frictionless customer experience of online shopping into the store wherever they can. They want access to the same type of rich data and high-performance analytics that retailers use to drive websites and mobile shopping trips. Their goal is to have that same limitless control to craft a customer experience and collect detailed data to help them predict how customers will shop.

The differentiation with IoT will come from a retailer's ability to sense, understand and act on IoT data with analytics. It won't be in the technology, the devices or the IoT plumbing. To take advantage of this new promising area, retailers should focus on IoT applications that better serve customers and create value.

Note that RFID works better than Barcodes. A significant advantage of RFID devices over the others mentioned above is that the RFID device does not need to be positioned precisely relative to the scanner. We're all familiar with the difficulty that store checkout clerks sometimes have in making sure that a barcode can be read. And obviously, credit cards and ATM cards must be swiped through a special reader.

In contrast, RFID devices will work within a few feet (up to 20 feet for high-frequency devices) of the scanner. For example, you could just put all of your groceries or purchases in a bag, and set the bag on the scanner. It would be able to query all of the RFID devices and total your purchase immediately. (Read a more detailed article on RFID compared to barcodes).

RFID technology has been available for more than fifty years. It has only been recently that the ability to manufacture the RFID devices has fallen to the point where they can be used as a "throwaway" inventory or control device. Alien Technologies recently sold 500 million RFID tags to Gillette at a cost of about ten cents per tag.

One reason that it has taken so long for RFID to come into common use is the lack of standards in the industry. Most companies invested in RFID technology only use the tags to track items within their control; many of the benefits of RFID come when items are tracked from company to company or from country to country.

However, common problems with RFID that are worth to pay attention to are *reader collision* and *tag collision*. Reader collision occurs when the signals from two or more readers overlap. The tag is unable to respond to simultaneous queries. Systems must be carefully set up to avoid this problem. Tag collision occurs when many tags are present in a small area; but since the read time is very fast, it is easier for vendors to develop systems that ensure that tags respond one at a time.

4.5.5.5. Cloud Ramifications

Spark's utility extends beyond the Internet of Things and its sophisticated transactions and includes a number of repercussions for the Cloud as well, which has become the de facto home for Big Data applications. Most of its IoT transaction capabilities can apply to general Cloud deployments where its advantages of combining graph analytics, SQL, and sensor data processing or streaming are most useful—especially with time-sensitive data requirements. The nexus point between IoT transactions and the Cloud that Spark can play an integral role in is the expansion of Clouds to what amount to IoT Clouds. Labeled as 'fogs', IoT clouds help to create a situation in which a Cloud's core resources (such as bandwidth and storage) are decentralized and pushed to its extremities in order to aggregate these resources across Clouds. In the case of the IoT, all of the interconnected devices will have their own Clouds that are effectively aggregated to conserve resources and reinforce interconnectivity—which

will require the sort of Machine Learning algorithms and graph analytics that Spark can readily provision.

4.5.5.6. Data Streaming

Another vital aspect of Spark is its tools for data streaming and accommodating continuously generated machine or sensor data. Although there is no shortage of options for streaming data, one of the distinguishing points of Spark's tool is that it processes events in a batch-like method during brief intervals of time in which events are continuously collected. Although Spark's streaming capabilities represent just one of many options of this technology that is still maturing, it helps to provide a gestalt of sorts with its other non-streaming capabilities (discussed above). Thus, the platform ensures that it can still enable the sort of speed necessary for real-time analytics on Big Data sets that are nimble enough for transactions in an Internet of Things world.

4.5.5.7. Under the Hood

At the abstraction level, Spark consists of a number of collections known as Resilient Distributed Datasets (RDD), which are intractable and based on local files (such as HDFS or others). There are several collection operations common to solutions written in Scala that apply to RDD, including parallelized versions of for each and map. Other functions include capabilities for reduce-by which consolidates entries according to a certain function pertaining to a specific key, and join capabilities that collect entries from a pair of RDD based on a shared key. Of particular use is the fact that Spark internalizes the operation sequence that resulted in a particular set of data, and can reconstruct that sequence in the event of node failure.

4.5.5.8. Dynamic Transactions

Given its propensity for enhancing Hadoop and the ubiquity of this Big Data platform, it is unlikely that the upstart Spark will replace Hadoop in the near future. Its assortment of tools, however, will help to facilitate an environment in Data Management where vendors continue to push the envelope for real-time analytics that exploit many of the aspects of Big Data (Machine Learning, graph analytics, streaming, continued SQL support, multiple development languages, and in-memory technologies) that will ensure that this technology has continued relevance. Redefining the nature of transactions, particularly in the impending wake of the Internet of Things, may be the best way for ensuring that failing Big Data initiatives do not contribute to this technology's untimely demise. Dynamic transactions based on all data, real-time and otherwise, can impact the enterprise's ROI in a number of ways from marketing to improved customer relations.

4.5.6. Natural Language Process Involvement

There are pivotal aspects of both Machine Learning and Cognitive Computing that are predicated on Natural Language Processing (NLP). Although IBM's Machine Learning APIs are available exclusively through Bluemix and various offerings related to Watson, they help to highlight some of the critical ways in which text analytics via NLP can create a cognitive

focus for application building. Some of the more utilitarian Machine Learning services involving NLP include capabilities for relationship extraction and user modeling. The former plies through sentences to identify various points of significance (subjects, actions, places); the latter creates predictions based on text and language analysis about social traits for specific people. Other NLP applications translate different languages and make sense of colloquial expressions. Although these particular features are associated with Watson, additional MLaaS providers have NLP features to enhance their offerings as well.

4.5.7. The Epicenter

In many ways, Machine Learning functions at the epicenter for a number of different facets of Big Data analytics. Its pivotal role only increases with the availability of MLaaS, which helps to democratize this subset of predictive analytics and enhance the roles of laymen and experts alike. As one of the enablers of the IoT, Machine Learning has a secure place in the future of Big Data. Its capacity to create timely action from analytics makes it essential to Big Data applications. Machine Learning's ultimate value for Big Data and analytics is alluded to in the aforementioned IBM blog:

"In many ways, MLaaS can be the Return-On-Investment (ROI) capstone of Big Data initiatives because Machine Learning algorithms can grow to be highly effective at data scales in volume, velocity and variety. Without MLaaS capabilities that can dynamically respond to myriad concurrent data streams in the cloud, the human race risks drowning in its own Big Data."

4.6. THE MATHEMATICS OF DATA: GRAPH ANALYTICS-AS-A-SERVICE

Graph databases are optimal for running advanced analytics because they indicate the relationship between data elements and allow for readily discernible inferences between them—yielding answers to questions that users never thought to ask. Leveraging the prowess of graph analytics (especially on Big Data sets), however, has traditionally been hampered by:

- A general lack of skills on any number of Semantics technologies (Sparql, RDF, ontologies, etc.) required to get the sort of performance needed for business cases.
- A lack of financial resources to provide the sort of heavy spending on infrastructure that some of the most eminent IT departments—such as those at Google and Facebook—readily utilize for their lucrative deployments.
- A dearth of analytics expertise regarding the most viable means of producing the sort of results needed to derive business value from data.

Graph Analytics as a Service (GAaaS), however, can rectify virtually all of these impediments while contributing these additional boons:

- Data integration in a single repository that involves all types of data regardless of models and schema, yet which is able to best other Data Lake options by providing much valued semantic consistency.
- Reinforcement of the self-service movement in Data Management that simplifies many processes (especially pertaining to analytics) and empowers business users.
- Democratizing semantic technologies while spurring adoption rates in a way that begets in earnest the trend towards ubiquitous computing and its importance in the coming years.

According to Algebraix Data CEO Charlie Silver, modestly priced Graph Analytics as a Cloud service (as compared to on-premise deployments and their considerable infrastructure requirements) ultimately helps level the playing field and gives small- and mid-sized businesses the same resources and advantages of large enterprises:

"Larger companies have tens of thousands databases whereas small companies have dozens of databases and they are all siloed," Silver said. "Setting up a linked data graph helps get rid of these silos and simultaneously allows for you to do deep reasoning inference analytics."

Figure 4-8. Network of Graph Database.

4.6.1. Deep Reasoning Inference Analytics

Graph databases provide an environment in which users can determine logical inferences about their data in ways that relational databases cannot. Graph analytics then exploit those inferences to make predictions and prescriptions about data elements in a reasonable way that acknowledges relatively simple relationships between data, which can make for much more profound analyses. Those simple relationships can become more revealing the more one utilizes this form of analytics. "As that graph has more data elements on it, the more inferencing and reasoning it can do" Silver said. "It's so rich and so full of insight that that's why we're taking this to the Business Intelligent (BI) analytics world directly."

4.6.2. Data Algebra

While the logical inferences of graph databases help to provision analytics based on overall views of data and their relationships to one another, those databases and analytics options are also immensely improved by a mathematical representation of data that makes graph analytics much more robust than other options. Algebraix Data is largely able to issue its service through the Cloud due to Data Algebra, in which both data and queries are represented as equations. By applying these mathematical representations to a triple store, the analytics vendor was able to achieve:

- Improved scalability to accommodate Big Data sets.
- Performance based rates of speed that issue results in close to real time and considerably expedite the deployment of critical technologies such as RDF and Sparql. The underlying engine for Algebraix Data can eschew conventional computations in any number of ways by utilizing algebraic representations and recognizable query patterns, or speed through them by re-using results and selecting optimum algorithms and data structures to perform them.
- Data integration regardless of sources, structure, data type, and format.

Although the aforementioned benefits of Data Algebra are significant, its potential to render graph databases as integration hubs may be the most salient advantage it offers—as well as the source of those other advantages:

> "We're going after this new market of graph analytics, but the ultimate play is going to be in data integration, where all data can integrate and officially apply to all enterprise data and the Internet," Silver remarked. "All data can be integrated irregardless of their model, which is a visual artifice because it can all be implemented and represented."

4.6.3. Integration, Data Modeling, and Data Lake Architecture

At the core of the integration capabilities that Algebraic Data drives and the celeritous, reliable results it delivers for graph analytics is an environment in which all data appears as mathematics regardless of modeling specifications. This fact is attributed to its Data Lake architecture (which is typical of NoSQL options), and which actually encourages and rewards enterprises for the incorporation of as much data as possible on the linked enterprise database graph. Silver noted:

> "Algebra for data…means that data, regardless of its structure or format—meaning it could be in a relational table, a graph, it could be hierarchical, voice, video, it doesn't matter—can be represented mathematically or algebraically, which is a huge innovation in the history of software development because what it really allows for is data to arc integrative."

4.6.4. Governance and Self-Service Graph Analytics

As is the case with any Data Lake and the rapid integration of disparate sources, there are very real governance concerns which Silver notes are "customer specific" and must be addressed in order to sustain such an option. However, by offering graph analytics as a Cloud-based service, service providers can potentially mitigate these concerns by leveraging the knowledge and experience of their staffs to help ameliorate this (and most other) aspects of this analytics process. Customers can simply pay to have professionals provide semantic consistency for their various data types.

In fact, there are a number of similarities between GAaaS and developments in self-service Business Intelligence and analytics options in which laymen end users do little more than apply the results of their analyses to specific business processes. According to Silver, customers simply need to give their data (which is likely in a relational format) to the company which in turn handles the lengthy process of implementing it into a linked data graph. That process typically includes conducting Extract Transfer and Load (ETL) or requisite data transformations, putting data into a graph, and educating users about how to issue queries while applying ontologies. "Our key value proposition is this," Silver asserted. "We use our mathematics to make this stuff all easy to use; we're essentially bringing that to the customers – making semantic technologies or graph analytics easy to use."

4.6.5. Ubiquitous Computing

The reduced ease which GAaaS allows for those attempting to utilize semantic technologies should play a critical role in democratizing semantics and helping to level the playing field in the capabilities of large enterprises and small- to mid-sized operations. Nonetheless, Silver believes the future of semantics and graph databases firmly lies in the latter's integration capabilities—particularly as augmented by the mathematical representations that Algebraix Data utilizes. The forms those possibilities take vastly exceeds providing analytics for individual customers, and pertains to a future in which perhaps all data—the world's data—is united in a single repository. "The world's data is billions of silos; everything is a silo," Silver acknowledged. "How do you bring those data to be integrated? Now, when you have those common mathematics, that's going to be a big change. That's going to happen over 20, 30 years; its' not going to happen tomorrow."

Such integration would account for ubiquitous computing, which would enable access from anywhere with any variety of devices and provide a degree of interoperability (between data types, organizations, countries, use cases) that is currently not available. Silver is convinced that the method of mathematically representing data that his company employs will ultimately be the enabler of such computing:

> "The mathematics of data has implications everywhere in the data world; everywhere in IT," Silver said. "However, we are narrowly focused right now on the Graph Analytics as a Service. Over time, our intellectual property will be the foundation for many, many applications that I can't even begin to describe."

4.7. CLOUD DATABASE

A cloud database is a collection of content, either *structured* or *unstructured*, that resides on a private, public or hybrid cloud computing infrastructure platform.

Two cloud database environment models exist: traditional and DataBase as a service (DBaaS).

In a traditional cloud model, a database runs on an IT department's infrastructure via a virtual machine. Tasks of database oversight and management fall upon IT staffers of the organization.

Note that Virtual Machine (VM) is an Operating System (OS) or application environment that is installed on software, which imitates dedicated hardware. The end user has the same experience on a virtual machine as they would have on dedicated hardware.

4.7.1. What is Database as a Service (DBaaS)

By comparison, the DBaaS model is a fee-based subscription service in which the database runs on the service provider's physical infrastructure. Different service levels are usually available. In a classic DBaaS arrangement, the provider maintains the physical infrastructure and database, leaving the customer to manage the database's contents and operation.

Alternatively, a customer can set up a managed hosting arrangement, in which the provider handles database maintenance and management. This latter option may be especially attractive to small businesses that have database needs, but lack the adequate Information Technology (IT) expertise.

4.7.2. Database as a Service (DBaaS) Benefits

In addition to the benefits of employing a cloud database environment model, contracting with a DBaaS provider offers additional benefits:

- **Instantaneous scalability:** Should added database capacity be necessitated by seasonal business peaks or unexpected spikes in demand, a DBaaS provider can quickly offer additional fee-based capacity, throughput and access bandwidth via its own infrastructure. A database operating in a traditional, on-site infrastructure would likely need to wait weeks or months for the procurement and installation of additional server, storage or communications resources.
- **Performance guarantees:** Through a service level agreement (SLA), a DBaaS provider may be obligated to provide guarantees that typically quantify minimum uptime availability and transaction response times. An SLA specifies monetary and legal remedies if these performance thresholds are not met.

Specialized expertise: In a corporate IT environment, except for the largest multinational enterprises, finding world-class database experts may be difficult, and keeping them on staff

may be cost prohibitive. In a DBaaS environment, the provider may serve thousands of customers; thus, finding, affording and keeping world-class talent is less of a challenge.

- **Latest technology:** To remain competitive, DBaaS providers work hard to ensure that all database software, server operating systems and other aspects of the overall infrastructure are kept up to date with security and feature updates regularly issued by software vendors.
- **Failover support:** For a provider of database services to meet performance and availability guarantees, it is incumbent on that provider to ensure uninterrupted operation should the primary data center fail for any reason. Failover support typically encompasses the operation of multiple mirror image server and data storage facilities. Handled properly, failover to a backup data center should be imperceptible to any customer of that service.
- **Declining pricing:** With advances in technology and an intensely competitive marketplace among major service providers, pricing for a wide range of cloud-computing services undergoes continual recalibration. Declining prices are a major impetus for migrating on-site databases and other IT infrastructure to the cloud.

4.7.3. Cloud Database Architecture

Cloud databases, like their traditional ancestors, can be divided into two broad categories: relational and nonrelational.

A relational database, typically written in Structured Query Language (SQL), is composed of a set of interrelated tables that are organized into rows and columns. The relationship among tables and columns (fields) is specified in a schema. SQL databases, by design, rely on data that is highly consistent in its format, such as banking transactions or a telephone directory. Popular choices include MySQL, Oracle, IBM DB2 and Microsoft SQL Server.

Nonrelational databases, sometimes called NoSQL, do not employ a table model. Instead, they store content, regardless of its structure, as a single document. This technology is well-suited for unstructured data, such as social media content, photos and videos.

4.7.4. Cloud Database Benefits

Compared with operating a traditional database on an on-site physical server and storage architecture, a cloud database offers the following distinct advantages:

- **Elimination of physical infrastructure:** In a cloud database environment, the cloud computing provider of servers, storage and other infrastructure is responsible for maintenance and availability. The organization that owns and operates the database is only responsible for supporting and maintaining the database software and its contents. In a DBaaS environment, the service provider is responsible for

maintaining and operating the database software, leaving the DBaaS user responsible only for his own data.

- **Cost savings:** Through the elimination of a physical infrastructure owned and operated by an IT department, significant savings can be achieved from reduced capital expenditures, less staff, decreased electrical and Heating, Ventilation, and Air Conditioning (HVAC) operating costs, and a smaller amount of needed physical space.

4.7.5. Migrating Legacy Databases to the Cloud

An on-premises database can migrate to a cloud implementation. Numerous reasons exist for doing this, including the following:

- Allows IT to retire on-premises physical server and storage infrastructure;
- Fills the talent gap when IT lacks adequate in-house database expertise;
- Improves processing efficiency, especially when applications and analytics that access the data also reside in the cloud; and
- Achieves cost savings through several means, including:
  - Reduction of in-house IT staff;
  - Continually declining cloud service pricing; and
  - Paying for only the resources actually consumed, known as pay-as-you-go pricing.

Relocating a database to the cloud can be an effective way to further enable business application performance as part of a wider software-as-a-service deployment. Doing so simplifies the processes required to make information available through internet-based connections. Storage consolidation can also be a benefit of moving a company's databases to the cloud. Databases in multiple departments of a large company, for example, can be combined in the cloud into a single hosted database management system.

4.7.6. How does a Cloud Database Work?

From a structural and design perspective, a cloud database is no different than one that operates on a business's own on-premises servers. The key difference lies in where it resides.

REFERENCES

[1] http://www.dataversity.net/expanding-master-data-management-with-big-data/.

Chapter 5

Artificial Intelligence Systems and Robots of Tomorrow

Talk about today's Artificial Intelligence (AI) and Robotics of Today versus what they need to be tomorrow, More Autonomous and independent from human and man in the loop. The convergence of Robotic Process Automation (RPA), machine learning, cognitive computing, artificial intelligence, and advanced analytics are driving unparalleled business model transformation. In this uncharted territory, enterprises need a partner that can help them seamlessly integrate people and machines, while simultaneously harnessing the technological disruption into competitive advantage.

5.1. Introduction

Artificial Intelligence (AI), the theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision making, and translation between languages. As part of Cognitive Science (CS), which is pioneering interdisciplinary field of studies between the nature of intelligent systems, both *Biological* and *Artificial*. See Figure 5-1.

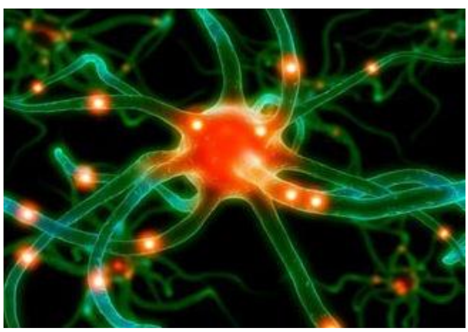


Figure 5-1. Biological Illustration of Neural Structure.

Drawing from the fields of computer science, linguistics, philosophy, neuroscience, and psychology, with strong links to anthropology, education, physics, and engineering, this field

of inquiry focuses on topics extending from accounts of human language and childhood mental development, to theories about how neurons in the brain process information; from attempts to model human thought in computer programs and to engineer robotic 'creatures' that live and learn, to studies of human vision and other forms of perception.

The science of cognitive and artificial intelligence are going hand to hand and we said they inter-mingle with each other. Artificial intelligence could match and then surpass human intelligence. Corporations and government agencies around the world are pouring billions into achieving AI's Holy Grail—human-level intelligence. Once AI has attained it, scientists argue, it will have survival drives much like our own. We may be forced to compete with a rival more cunning, more powerful, and more alien than we can imagine.

Although, profiles of technical visionaries, industry watchdogs, and groundbreaking AI systems, our final Invention explores the perils of the heedless pursuit of advanced AI. Until now, human intelligence has had no rival and the questions are:

1. Can we coexist with beings whose intelligence dwarfs our own?
2. And will they allow us to?

For example, what are the robots are used in automobile manufacturing and other industries in today's world, versus what they need to be doing in future, to take over tasks that human brain can do, in order to expand on this matter. Therefore, we need to understand certain aspect of computers technology of past, present and future as well as how the new upcoming biochip impact the futuristic infrastructures of these computer as well as influence of neural networking in design of these computers.

As David L. Anderson states [1], what is the relationship between computers and human beings? Whether or not humans are essentially computers, as some theories assert, learning does involve "information processing." Some educational methods (computer-based and otherwise) for example, require students to handle information in a mechanical way that undermines both the development of critical skills and a genuine understanding of the material. This essay is a reflection on the ways in which computers in education.

In Chapter 3 we covered basics of neural networking and its relation to Artificial Neural Net (ANN) as well, we even construct neural networks to model a simple binary logical functions by starting with feed-forward networks, encounter the XOR logical problem, and solve it by introducing the concept of backpropagation. The module was consistent of a working back propagating neural net capable of solving any binary logical function as simple as 1 + 1 = 2 situation as it is illustrated in Figure 5-2:

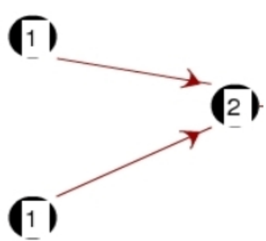


Figure 5-2. Logical Binary of Computer.

The modular components of Artificial Neural Net (ANN) can be summarized and described as follows:

- **Introduction to Neural Networks**
  Neural network in near future from design and fabrication point of view based on cognition science will be in position to act possibly near the human brain. They may composed of massive numbers of small units that are connected together in interesting ways. In this module, we discover the basic structure of neural networks, and how these simple networks are able to realize basic computational processes.
- **Simple Neural Nets for Logical Function**
  Simple feed-forward neural nets can be arranged to model a number of different simple logical functions, such as 'and', 'or' and 'majority'. In this format, one will has a chance to create such networks to solve functions of his or her choice
- **The XOR Problem and Solution**
  Simple networks have two drawbacks: they depend on architectural design set by a programmer, and they cannot solve for discontinuous functions like 'XOR'. In this module, we introduce networks that can correct their own architecture through back propagating error signals

Beyond what we have describe in Chapter 3 and here in this chapter readers can refer to a site set up by Anderson for further analysis and discussions [2].

The idea of implementing Artificial Intelligence is nothing new, as matter of the AI technology falls back into more than 20 years ago when it was first thought of as part battle management of Strategic Defense Initiative (SDI), known as Star Wars. Although the first generation of these AI were not as dynamic as we want them going forward but they were revolutionary given the period of time and technology of 20 years or so ago. It was required that the databases and information known to human being to be on board of consultation satellites that were using these AIs with precise information of the threats against them to be able to use these AIs to defend themselves by taking proper counter-measure against measure of threats and utilized the capabilities of their battle managements on board. See the book by Zohuri [3] for further information.

However with flow of data coming from every directions at speed of network communication passing through network as result flow of information and its processing requires a new generation of AIs systems that pretty much they can be on their own, based on Fuzzy Logic (FL) approach rather than Boolean Logic (BL). Therefore, they can process the right information in real time and pass them to right stakeholder for final decision making where the proper actions can be, taken.

Furthermore, the infrastructure of these smarter AIs cannot be built on old technology of the computer systems that we have on our desktop or lap top. Thinking in Silicon need to be taken in totally different direction and creation of new Bio Chip need to be in order, thus our approach to the new generation of computers should go beyond classical one to something possibly known as non-classical one.

5.2. THE NATURE OF ARTIFICIAL INTELLIGENCE

AI (pronounced AYE-EYE) or Artificial Intelligence is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using the rules to reach approximate or definite conclusions), and self-correction. Particular applications of AI include expert systems, speech recognition and machine vision.

AI invention was the brainstorm of John McCarthy, an American computer scientist at Dartmouth Conference, in 1956, where the discipline was born. Today, it is an umbrella term that encompasses everything from robotic process automation to actual robotics. It has gained prominence recently due, in part, to big data, or the increase in speed, size and variety of data businesses are now collecting. AI can perform tasks such as identifying patterns in the data more efficiently than humans, enabling businesses to gain more insight out of their data.

AI can be, categorized in any number of ways, but there are two examples that we can discuss here in this section. The first classifies AI systems as either ***weak*** AI or ***strong*** AI and they are listed as follows:

1. Week Artificial Intelligence

The week AI also known as *Narrow* AI, is an AI system that is, designed and trained for a particular task. Virtual personal assistance, such as Apple's Sire on IPhone, are a form of week AIs. Narrow AI is an application of artificial intelligence technologies to enable a high-functioning system that replicates – and perhaps surpasses -- human intelligence for a dedicated purpose. Many current systems can be, classified as narrow AI. One well-known example is IBM’s Watson supercomputer, which applies cognitive computing, machine learning and natural language processing to perform as a “question answering” machine. Watson actually out-performed human contestant Ken Jennings to become the champion on the popular game show, Jeopardy! Essentially, Watson is a type of expert system, a computer program that uses AI technologies to simulate the knowledge and cognitive ability of a human within a particular realm. Subsequent expert systems based on Watson include an artificially intelligent attorney and a medical research assistant.

Most narrow AI applications are much less, sophisticated than Watson. Any software that uses technologies like machine learning, data mining, pattern recognition and natural language processing to autonomously, make decisions can be considered narrow AI. As such, narrow AI systems include spam filters, self-driving cars and Facebook’s newsfeed.

Narrow AI is also known as weak AI, in contrast with strong AI (which is also known as artificial general intelligence). Strong AI involves a system with comprehensive knowledge and cognitive capabilities such that its performance is indistinguishable from that of a human, although its speed and ability to process data is far greater. Such a system has not yet, been developed, and expert opinions differ as to the possibility that it ever might be.

2. Strong Artificial Intelligence

Strong AI also known as Artificial General Intelligence (AGI), is the representation of generalized human cognitive abilities in software so that, faced with an unfamiliar, task, it has enough intelligence to find a solution. The Turing Test, developed by mathematician Alan

Turing in 1950, is a method used to determine if a computer can actually think like a human, although the method is controversial.

A socially aware general-purpose AI could scan the web for information and, by reading human facial expressions, it could deliver targeted speeches better than any political leader, said Armstrong. Taken to the extreme, he warned that it is difficult to specify a goal that is safe: "If it were programmed to prevent all human suffering, the solution could be to kill all humans."

The second example is from Arend Hintze, an assistant professor of integrative biology and computer science and engineering at Michigan State University. He categorizes AI into four types, from the kind of AI systems that exist today to sentient systems, which do not yet exist. His categories are as follows:

- **Type 1: Reactive machines**. An example is Deep Blue, the IBM chess program that beat Garry Kasparov in the 1990s. Deep Blue can identify pieces on the chess board and make predictions, but it has no memory and cannot use past experiences to inform future ones. It analyzes possible moves -- its own and its opponent -- and chooses the most strategic move. Deep Blue and Google's AlphaGO were designed for narrow purposes and cannot easily be applied to another situation.
- **Type 2: Limited memory**. These AI systems can use past experiences to inform future decisions. Some of the decision making functions in autonomous vehicles have been designed this way. Observations used to inform actions happening in the not-so-distant future, such as a car that has changed lanes. These observations are not stored permanently.
- **Type 3: Theory of mind**. This is a psychology term. It refers to the understanding that others have their own beliefs, desires and intentions that impact the decisions they make. This kind of AI does not yet exist.
- **Type 4: Self-awareness**. In this category, AI systems have a sense of self, have consciousness. Machines with self-awareness understand their current state and can use the information to infer what others are feeling. This type of AI does not yet exist.

Bottom line is that artificial intelligence is marching in path to simulate the human intelligence by machine, it is just matter of time and technologies.

5.3. THE NATURE OF FUNCTIONALISM

Functionalism is a popular theory of the nature of minds. While there continues to be great controversy about which is the correct "theory of mind," functionalism is probably the most widely held theory among both scientists and philosophers today. On this theory, mental states (beliefs, pains, hopes, fears, etc.) are ultimately, characterized by the jobs they do, which is to say the functions that they perform. Since computers just are mechanical devices that implement functions, this makes the computer metaphor a natural way of capturing the main intuitive idea behind the theory. On this account, our brains are like the hardware of a computer and our minds (our beliefs and pains) are like the software states of a computer.

This section allows to understand analogies to help explain the theory of functionalism, why it is a compelling theory to use in scientific research, and why it raises such passionate resistance.

Note that: Functionalism is also sometimes called the computational theory of the mind, although others reserve the computational theory to refer to one narrow version of functionalism.

5.3.1. Introduction to Functionalism

Here, we directly quote David Anderson and his introduction to Functionalism as reference [4].

For several thousand years, philosophers and theologians have speculated about the nature of the human mind. Many fascinating theories have been, advanced to give an account of the essence of mental states. Throughout most of that history, the dominant views have assumed that the mind is something quite mysterious and fundamentally unlike anything else in the natural world. On that assumption, the mind would not be the kind of thing that could be studied with the methods of science.

It is only in the past century that attempts have been, made to develop a “science” of the mind. For a theory of the mind to be scientific, it must be empirical. That is, it must give an account of the mind in terms of properties that are accessible to the five senses (sight, hearing, taste, touch and smell). If a theory of this kind can indeed capture the fundamental nature of the mind and cognition, then the mind would no longer be an inscrutable mystery, it could be, studied just like anything else in nature. It would, as we say, to be “publicly accessible.”

In this section, we will explore the most influential contemporary theory of the mind: functionalism. It is an empirical theory that has spawned a massive research effort that will take place, if it fulfills its promise, explain much about the mind that presently remains beyond our understanding. While functionalism dominates present research in the cognitive sciences, it remains a controversial theory. There exists a vocal minority, who insist that functionalism fails to capture essential elements of “the mind.” Later, we will examine some of the most famous arguments against functionalism. But for now it is worth noting that even those who reject the theory sometimes admit that there is no better empirical theory presently on the horizon. These same people often agree that research programs built upon the assumptions of functionalism are enormously useful. Even if functionalism is eclipsed or amended some day by a new theory, it is likely that any new theory will have benefited from the lessons learned by research programs that push functionalism to its limits. And so now we turn to functionalism, the most influential contemporary theory of the mind.

5.3.2. Heating Systems and Functions, of Mousetraps

If you see a machine that you have never seen before, you might ask: “*What is that contraption*? *What is its purpose*? *What is it for*?” In asking those questions, you are asking for an explanation of the function that the machine serve. In many contexts, when we ask, “What is it?” what we ultimately are asking for is an account of what it does.

Consider, for example, a mousetrap. If you know that someone has purchased a mousetrap, what you know is the function it performs. It is designed (either well or badly) to capture mice. There are dozens of types of mousetraps, from the familiar spring-loaded variety with cheese, to poisonous chemicals placed in a “mouse hotel.” Knowing only that it is a mousetrap, you know nothing about how it performs its function or what materials it is, made of. And for that, you may not care -- so long as it performs its function well. It is instructive that an expression has entered our language that focuses on the purely functional properties of mousetraps. If someone is trying to create a new invention, a new method of performing a familiar function, we say, “they are trying to build a better mousetrap.”

We can say the concept of a mousetrap can be ***realized*** (that is, made into a specific, tangible object) according to many different designs and using many different materials. To borrow an expression from cognitive science, we can say that mousetraps can be **multiply, realized** in different physical contraptions. As we will come to learn, if functionalism is true, then minds can be, multiply realized in different designs and different materials as well. However, more details about that will come later [4].

5.3.3. A Functional Description of Heating System

Let us consider the function of an automated heating system. Its primary function is to produce heat whenever the air temperature drops below a predetermined level. We can say that the system takes as input the air from the room and gives as output heat, as needed. If we assume that the heating system has been set for 68 degrees Fahrenheit, then we can say that its function is to produce heat if the temperature is below 68 degrees F, and to do nothing if the temperature is 68 degrees F or higher. As represented in the image below, it takes the air from the room as input, and it gives heat (or not) as output according to a rule [4].

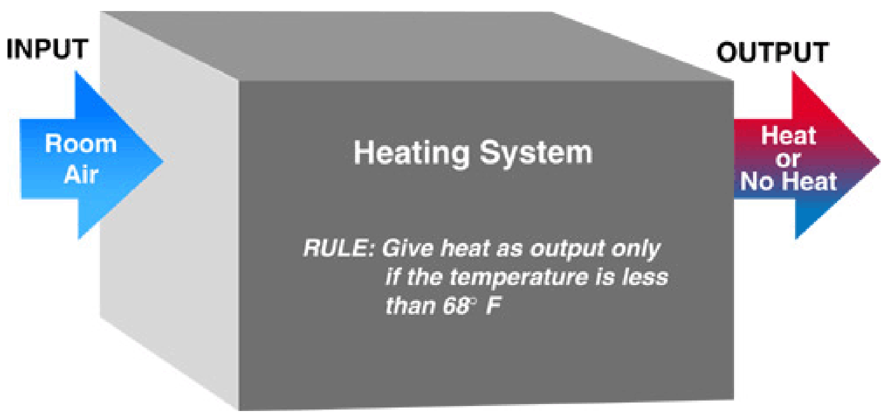


Figure 5-3. Convectional Heat Transfer in a Room.

Here, we are treating the entire heating system as a black box -- that is to say, we ignore what is inside the box. We are concerned only with the behavior of the system, not with its construction or the internal workings of its parts. Treating it as a black box we know only what its output is (i.e., how it behaves) in response to various inputs. Upon careful

observation of its behavior, we would discover that this heating system implements the following function (or rule) [4]:

Give heat as output *only if* the temperature is less than 68 degrees

If we get curious, though, and want to "peek inside" the black box, we will find whatever physical components are, used to perform the functions of this particular automated heating system. For example, we might find the following:

As we examine the components that actually make the heating system function properly, we will find that there are several distinct devices, each making its own unique contribution to the system. The system pictured above, and most heating systems, include (among other things) the three following components [4]:

1. A thermostat
2. A switch
3. A heating source

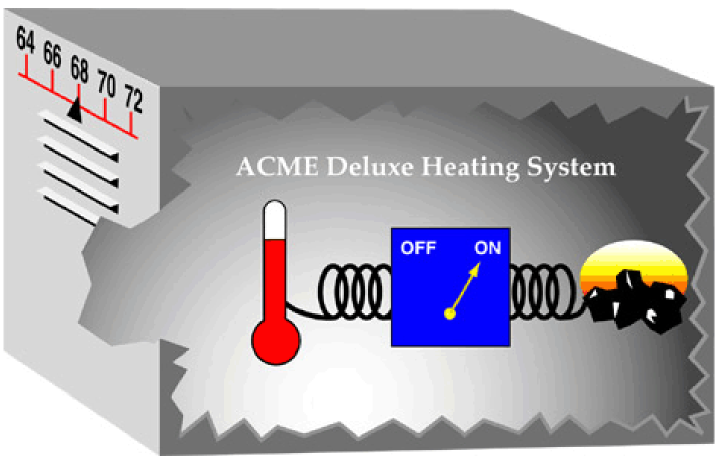


Figure 5-4. Illustration of Black Box of Subject.

Let us take the first device, the thermostat. If we are interested in how the device is actually constructed, we can examine it so closely as to be able to write complete blueprints that would enable us to reproduce the device in every detail. Such blueprints would identify the dimensions of each separate component of which it is constructed, it would specify the materials used to construct each component, and it would show precisely how all of the parts are arranged so as to operate properly. In the case of the heating system pictured above, we see that something like a familiar mercury (or red alcohol) thermometer is at the heart of the thermostat. A detailed blueprint would tell us what kind of glass the tube is made of, the shape of the glass, the volume of mercury enclosed within the glass, etc. A thermometer alone, however, does not a thermostat make. In addition to the mercury thermometer, there would have to be additional components capable of reading the temperature on the thermometer (a photoelectric sensor would work) and a device for determining whether the temperature recorded by the thermometer was or was not below the temperature on which the

thermostat was set. (We have left these other components out of our picture for simplicity sake) [4].

We might have an interest in the physical properties of the thermostat, but we might not. We might instead be interested solely in the thermostat's functional properties. As a way of expressing this shift in perspective, we can imagine a black box, which encloses the thermostat, hiding it from view. We can then do for the thermostat, what we had initially done for the entire heating system -- we can give a functional description of the thermostat, noting only what it takes as input and what it gives as output, ignoring completely how it succeeds in performing that function. The function of a thing is often called its causal role, that is, the effect that it has on other parts of the system (of which it is a part) and the effect that it has on the world around it. To speak of the function or the causal role of a thing, is not to speak of how it causes certain things to happen, but merely that it does so [4].

Just as we have given a functional description of the thermostat, we can do the same thing for two other key components of a heating system, the "switch" and the "heat source."

Each of the components of the heating system, above, is represented in functional terms not physical terms. We can see what goes into each black box and we can observe what output it gives in response to each input. If we pay attention, we may be able to discover the rule that it follows in giving a particular output for each input. For all this, we can see what it does, but not how it does it. Given that there are many ways to build a thermostat that performs the "temperature detecting" function, we say that a thermostat can be **realized** in many different ways [4].

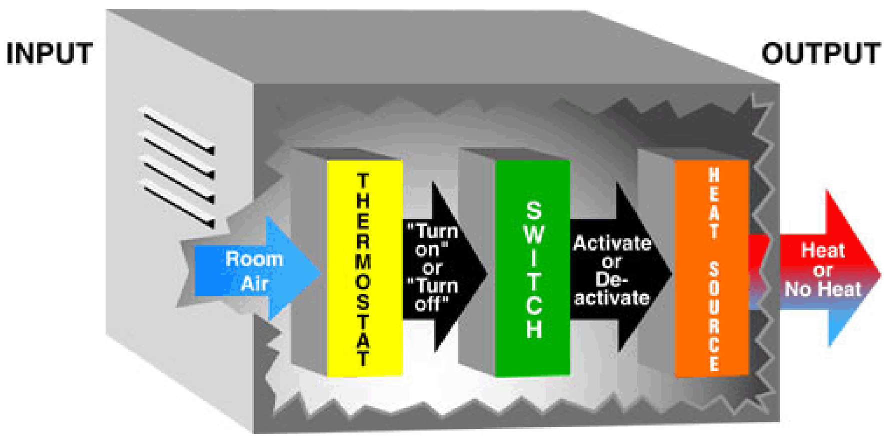


Figure 5-5. Heating System inside the Black Box.

There are many possible designs for a thermostat. At the heart of every thermostat is a thermometer, a device that can accurately detect the temperature of the air. Consider the three possible ways of building a thermometer as shown on the left-hand side of the image below. Each uses a different method to measure the air temperature. Any one of them will perform the same function, and so while their physical properties will be quite different, their functional properties will be identical.

Just as there are different ways of building thermostats, there are also different ways of building switches and heat sources. In the picture below, you can see three different ways of building (or *realizing*) each of the functional devices: thermostat, switch and heat source [4].

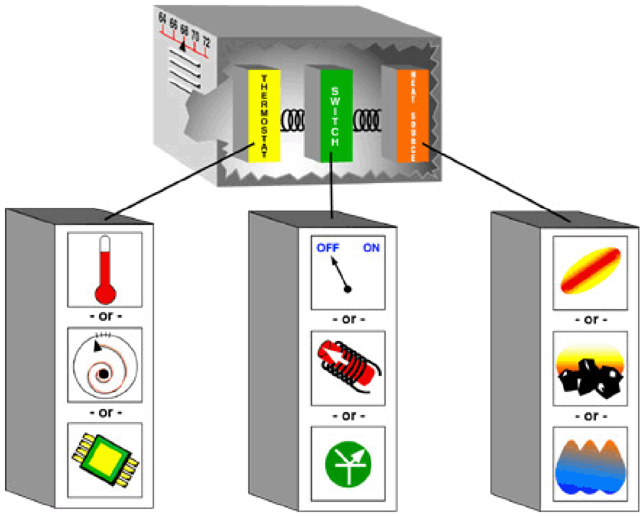


Figure 5-6. Conceptual Design of Thermostats.

If you want to build a heating system, you can pick any thermostat from column A, any switch from column B and any heating source from column C. Connect them all together (with a little electrical wire and duct tape) and you will have a heating system [4].

5.3.4. Back to the Functionalist Theory of the Mind

Functionalism is a theory of the mind that claims to tell us the fundamental nature of our mental states. Your mental states include everything from your fear of heights to your belief that it is Tuesday, from your desire to eat an ice cream cone to the sharp pain in your leg. Fears, beliefs, desire, pains are all mental states. Your mental states include the cognitive processing that you are aware of because they are, accompanied by subjective experiences, or “feelings,” of some kind. But they also include the cognitive processing that you are not aware of, like the way your visual system turns simple two-dimensional line-drawings into a three-dimensional “world” [4].

So, what does your desire for an ice cream cone and the pain in your leg have to do with thermostats and heating systems? Well, according to functionalism, the essential nature of your desires and your pains is not to be found in the stuff that your desires are composed of, but rather in the function that each performs. What kind of stuff is your pain made of? Are pains somehow made of molecules (i.e., physical stuff)? Or are they made of some kind of immaterial mental stuff. According to functionalism, we can describe the essential features of our mental states without mentioning what kind of stuff it is made of. We need only describe what goes into and what comes out of each of the component black boxes. An account of a particular mental state, like pain, is complete when all of its functional properties have been, identified. The stuff that is used to implement those functional properties is not an essential part of what mental states really are [4].

Now it must be, said that virtually all contemporary functionalists are physicalists. That is, they believe that everything that exists is ultimately built out of physical stuff (molecules, etc.) as described by our best theories in physics. Being physicalists, they do not believe in

the existence of immaterial, mental stuff. Further, even though a description of a mental state may not require any mention of the stuff of which it is, made, it may, nonetheless, prove enormously helpful to carefully examine the stuff if you want to discover precisely what function the stuff performs. (That is, if you seek a thorough account of the causal role of pain in a particular organism, examining the activity of nerve endings and pain centers in the brain may provide relevant data.) Nonetheless, even though functionalists tend to be physicalists, they must concede that it is at least imaginable that pain could be realized in something other than molecules; just as we all must concede that a thermostat could be realized using something other than a mercury thermometer. Functional properties can always be (at least in principle) multiply realized [4].

5.3.5. Functionalism, the Computer Metaphor and Extra-Terrestrial (ET)

In David Anderson discussion of functionalism thus far, the word “computer” has not been, mentioned. Yet, from the first time that functionalism was advanced as a theory of the mind (by the philosopher, Hilary Putnam, who is usually credited with being its first proponent in a paper called “Minds and Machines” 1960), it was explained using the computer metaphor. To say that mental states are functional properties, said Putnam, is to say that the mind is, in fundamental ways, like a computer program. Now, other people had already suggested that the mind was like a computer. In fact, the most popular theory of the mind at the time that functionalism was first introduced was a theory called the identity thesis. According to this theory, the mind was to be identified with the brain, and mental states were essentially brain states. Many who embraced this theory also appealed to the *computer metaphor*. They said that the brain was a computer and mental states were physical states of the brain/computer. However, there is a crucial difference between the identity thesis according to which mental states are states of the computer's hardware, and functionalism according to which mental states, are states of the computer's software. If you define 'pain' as a physical state, then no creature anywhere in the universe could be in pain unless it was in the same physical brain state that we are in when we are in pain. But that does not seem right. If you have seen the movie ET: The Extraterrestrial, you have seen an alien from another planet suffer to help save the life of a child. If we assume that the physical structure of ET's brain is quite different from ours (which it would likely be), then it couldn't be in the same physical brain state that we are in when we are in pain. If so, then, we would have to say that ET couldn't be in pain. But that doesn't seem right. Surely, it is possible for ET to be in pain, even if his brain is physically quite different from ours. (If you befriended ET like the boy in the movie, you would say that ET was in pain!) [4].

Here is one of the strengths of functionalism. Since mental states, like pains, are functional states rather than physical states, then pains can be multiply realized in a wide range of different physical states, and thus in all types of strange and wonderful creatures. Just as the term, 'thermostat' picks out a functional concept, so does 'pain'. If something fulfills the function of a thermostat in a heating system, if it has the causal role that we described above -- then it is a thermostat. It doesn't matter what it's made of. Likewise, if something fulfills the function of pain in an alien from another planet, if it has the same causal role in ET that it has in us -- then it is a pain. It does not matter how the alien's brain is designed or what it is made of. It does not even matter if ET has a unique organ that can be

described as a “brain.” It is still possible that the very same mental state implemented in us can also be implemented in ET. According to functionalism, ET can be in pain! [4].

So where does that leave us? Well, we know that many contemporary scientists and philosophers believe that your mind is very like a piece of software running on your brain, the hardware. But that only tells us so much. Do we really know what a computer is? We see them every day, of course. But you may be surprised that there is quite a debate among scientists as to what things are and what are not computers. Is your brain really a computer? In addition, if it is a computer, what kind of computer is it? The computer metaphor may offer a startling insight into the nature of our minds or it may be another theoretical dead end. However, until we get clear about what computers are, we can't even tell whether the computer metaphor offers an interesting theory that makes a controversial claim about the nature of our minds, or whether it is a trivial claim that says virtually nothing of interest about our minds. Therefore, our next stop must be an exploration of nature of computers. We think that you will find an exploration of computers more interesting than you might expect [4].

5.4. The Nature of Computers

Computers play a formative role in the scientific study of the mind and brain. One important research method is to model a particular mental ability within a computer program. Many researchers in the field go further and claim that the mind is literally a computer program running on your brain (See Section on Functionalism). However, what is a computer? That is a matter of considerable controversy itself.

What exactly is a computer? Computer is an electronic device that manipulates information, or data. It has the ability to store, retrieve, and process data. You may already know that you can use a computer to type documents, send email, play games, and browse the Web. You can also use it to edit or create spreadsheets, presentations, and even videos. Figure 5-7 is a simple calculator that presents the first generation of hand held computer.

What we see on a computer monitor are flashing lights and full color graphics. It is easy for observers to be, mystified by computers. This section describes the basic concept of a computer as a device that implements functions.

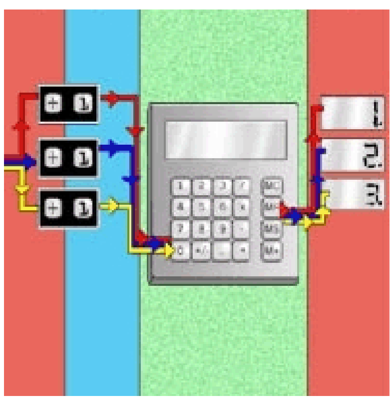


Figure 5-7. A Simple Classical Computer.

This section also uses simple analogies to explain how the same computer program can be implemented in wildly different kinds of material, and continues with a discussion of the classical digital computer, which is the type of computer that most of us have implemented and worked with in our daily lives.

It lays out the prominent features of digital computers (digital, serial, local, and deterministic) and provides a flash animation that gives a dynamic analogy to help drive home the main concepts. Discussion then turns to non-classical computers, computers that lack all or most of the essential features of digital computers and have instead some or all of the following properties: analog, parallel, distributed, non-deterministic. Some strengths and weaknesses of the two different types of computer are briefly, mentioned. This component does not currently include a discussion of the dynamical systems model of mental processing.

5.4.1. Hardware versus Software

Before we talk about different types of computers, let us talk about two things all computers have in common: ***hardware*** and ***software***.

1. Hardware: It is any part of your computer that has a *physical structure*, such as the keyboard or mouse. It also includes all of the computer's internal parts, which you can see in the image below as Figure 5-8.

Figure 5-8. Typical Mother Board of Classical Computer.

2. Software: It is any set of instructions that tells the hardware *what to do* and *how to do it*. Examples of software include web browsers, games, and word processors. Below, you can see an image of Microsoft PowerPoint as Figure 5-9, which is used to create presentations.

Everything you do on your computer will rely on both hardware and software. For example, right now you may be viewing this lesson in a web browser (software) and using your mouse (hardware) to click from page to page. As you learn about different types of computers, ask yourself about the differences in their hardware. As you progress through this tutorial, you will see that different types of computers also often use different types of software.

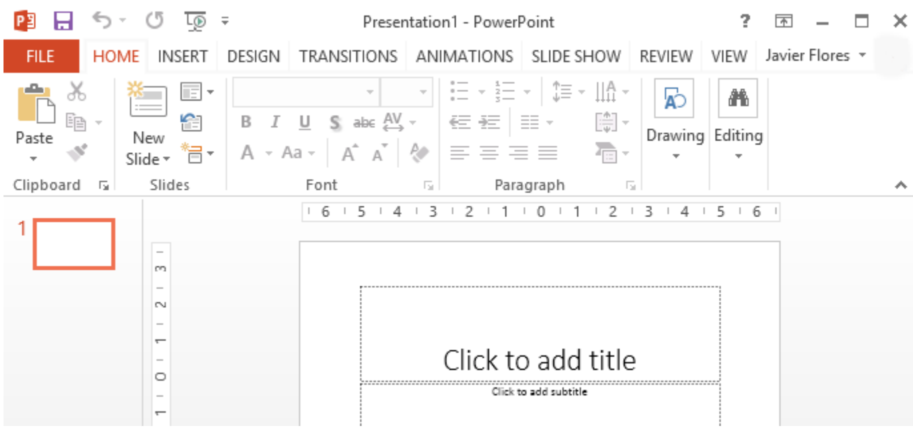


Figure 5-9. Illustration of Microsoft PowerPoint Screen.

5.4.2. Different Types of Computers

When most people hear the word computer, they think of a personal computer such as a desktop or laptop. However, computers come in many shapes and sizes, and they perform many different functions in our daily lives. When you withdraw cash from an ATM, scan groceries at the store, or use a calculator, you are using a type of computer.

- **Desktop Computers**

Many people use desktop computers at work, home, and school. Desktop computers are designed to be placed on a desk, and they're typically made up of a few different parts, including the computer case, monitor, keyboard, and mouse.

Figure 5-10. Illustration of Typical Desktop Computer.

- **Laptop Computers**

The second type of computer you may be familiar with is a laptop computer, commonly called a laptop. Laptops are battery-powered computers that are more portable than desktops, allowing you to use them almost anywhere.

Figure 5-11. Illustration of Typical Laptop Computer.

- **Tablet Computers**

Tablet computers—or tablets—are handheld computers that are even more portable than laptops. Instead of a keyboard and mouse, tablets use a touch-sensitive screen for typing and navigation. The iPad is an example of a tablet.

Figure 5-12. Illustration of a Typical Tablet Computer.

- **Servers**

A server is a computer that serves up information to other computers on a network. For example, whenever you use the Internet, you are looking at something that is stored on a server. Many businesses also use local file servers to store and share files internally.

Figure 5-13. Illustration of a Typical Server Computer Systems.

- **Other Types of Computers**
  Many of today's electronics are basically specialized computers, though we don't always think of them that way. Here are a few common examples.
  - Smartphones: Many cell phones can do many things computers can do, including browsing the Internet and playing games. They are often, called smartphones.
  - Wearables: Wearable technology is a general term for a group of devices—including fitness trackers and smartwatches—that are designed to be worn throughout the day. These devices are often, called wearables for short.
  - Game consoles: A game console is a specialized type of computer that is, used for playing video games on your TV.
  - TVs: Many TVs now include applications—or apps—that let you access various types of online content. For example, you can stream video from the Internet directly onto your TV.

5.4.3. PCs and Macs

Personal computers come in two main styles: PC and Mac. Both are fully functional, but they have a different look and feel, and many people prefer one or the other.

Figure 5-14. Image of a PC Computer.

This type of computer began with the original IBM PC that was, introduced in 1981. Other companies began creating similar computers, which were, called IBM PC Compatible (often shortened to PC). Today, this is the most common type of personal computer, and it typically includes the Microsoft Windows operating system.

Figure 5-15. Image of a Mac Computer.

The Macintosh computer was introduced in 1984, and it was the first widely sold personal computer with a Graphical User Interface, or GUI (pronounced gooey). All Macs are made by one company (Apple), and they almost always use the Mac OS X operating system.

5.5. COMPUTER TYPES: CLASSICAL VERSUS NON-CLASSICAL

This section introduces readers to the fundamental distinctions between *classical* and *non-classical* computers. Interactive flash animations help us grasp the fundamental differences between familiar digital computers and other computers that function differently.

In the previous section, the discussion focused on the classical concept of a computer, what we will often call a digital computer. It is the kind of computer that we are all most familiar with it. It is the kind that you use -- a Mac or a PC. It is the kind, which is, used in business and at school. In fact, virtually every computer that presently exists on the planet is a digital computer. Yet, there is another kind of computer that is, fundamentally different from the digital computer. It is so different, in fact, that some people insist that it should not even be called a computer. While we will eventually consider that question, we will begin by assuming that it is indeed a computer, just a different kind of computer.

As Professor Anderson put [1], while there are many names for it, we will usually call it a non-classical computer. But other names are used for computers that are not of the “classical” type: connectionist computer, artificial neural network, analog computer and parallel distributed processor, to name a few. The different names arise, in part, from an attempt to

identify a special feature that serves to distinguish it from digital computers. Since most non-classical computer possess several distinctly different features, it is not surprising that there are several names for it. We are now going to consider some of those properties. While not every, non-classical computer possesses all of these properties, we will assume that our typical non-classical computer (our "paradigm" of this type) does indeed have all of the properties described below.

Two of the properties that distinguish a non-classical computer from a digital computer involve the way that information is carried in the system. In a non-classical computer, at least some of the information is carried in analog rather than digital form and the information is distributed throughout large sections of the system rather than localized in specific places. For further information, readers need to refer to Anderson's website [5].

In summary, Non-classical logics (and sometimes-alternative logics) is the name given to formal systems that differ in a significant way from standard logical systems such as propositional and predicate logic. This is done, including by way of extensions, deviations, and variations in several ways. The aim of these departures is to make it possible to construct different models of logical consequence and logical truth.

Philosophical logic, especially in theoretical computer science, is understood to encompass and focus on non-classical logics, although the term has other meanings as well.

5.5.1. Digital (Not Analog)

Digital computer is falling into any of class of devices capable of solving problems by processing information in discrete form. These computers operate on data, including magnitudes, letters, and symbols that are expressed in binary format or code - i.e., using only the two digits 0 and 1 known to them. By counting, comparing, and manipulating these digits or their combinations according to a set of instructions held in its memory, a digital computer can perform such tasks as to control industrial processes and regulate the operations of machines. Analyze and organize vast amounts of any types of data from business to the others in very structured form; and simulate the behavior of dynamic systems (e.g., global weather patterns, security and cyber security, banking process related to its application or for that matter chemical reactions, etc.) in scientific research. Post-war, after brief competition from electronic based analogue computers, our familiar digital computer emerged from the designs of von Neumann and his contemporaries

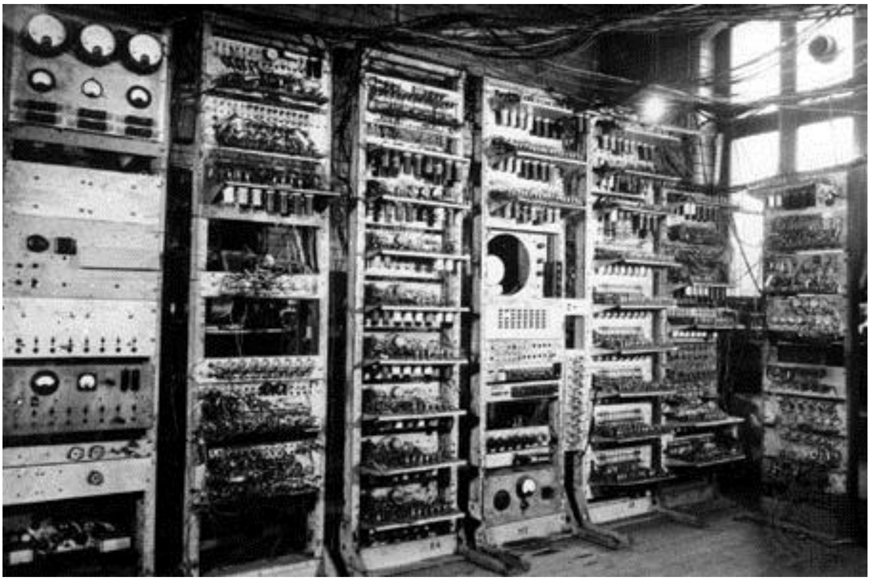
A typical digital computer system has four basic functional elements:

1. input-output equipment,
2. main memory,
3. control unit, and
4. arithmetic-logic unit

Any of a number of devices is, used to enter data and program instructions into a computer and to gain access to the results of the processing operation. Common input devices include keyboards and optical scanners; output devices include printers and monitors.

The information received by a computer from its input unit is stored in the main memory or, if not for immediate use, in an auxiliary storage device. The control unit selects and calls

up instructions from the memory in appropriate sequence and relays the proper commands to the appropriate unit. It also synchronizes the varied operating speeds of the input and output devices to that of the Arithmetic-Logic Unit (ALU) so as to ensure the proper movement of data through the entire computer system. The ALU performs the arithmetic and logic algorithms selected to process the incoming data at extremely high speeds—in many cases in nanoseconds (billionths of a second). The main memory, control unit, and ALU together make up the Central Processing Unit (CPU) of most digital computer systems, while the input-output devices and auxiliary storage units constitute peripheral equipment.



Reprinted with permission of the Department of Computer Science, University of Manchester, Eng.

Figure 5-16. The Manchester Mark I, the first stored-program digital computer, c. 1949.

Figure 5-10 here is the first generation of digital computer that was known as the Manchester Mark I that stored program digital format in 1949.

5.5.2. Analog (Not Digital)

An analog computer is a form of computer that uses the continuously changeable aspects of physical phenomena such as electrical, mechanical, or hydraulic quantities to model the problem being solved. In contrast, digital computers represent varying quantities symbolically, as their numerical values change. As an analog computer does not use discrete values, but rather continuous values, processes cannot be reliably, repeated with exact equivalence, as they can with Turing machines. Unlike digital signal processing, analog computers do not suffer from the quantization noise, but are, limited by analog noise.

Analog computers were widely used in scientific and industrial applications where digital computers of the time lacked sufficient performance. Analog computers can have a very wide range of complexity. Slide rules and monographs are the simplest, while naval gunfire control computers and large hybrid digital/analog computers were among the most complicated [6].

Systems for process control and protective relays used analog computation to perform control and protective functions.

The advent of digital computing and its success made analog computers largely obsolete in 1950s and 1960s, though they remain in use in some specific applications, like the flight computer in aircraft, and for teaching control systems in universities. Figure 5-11: is a typical analog computer illustration.

Figure 5-17. Illustration of an Analog Computer.

This analog feature makes our non-classical computer more resilient and less susceptible to breakdown than our digital computer, at least in one respect. A computer that has this property is often described as being *robust*; a computer that lacks this property and is easily defeated by minor malfunctions is typically described as *brittle.*

Note that the analog computers can perform much better in systems that are handling the battle management process of tomorrows star wars type battles due to the fact that they can sustain any Electro Magnetic Pulse (EMP) or Elector Magnetic Interference (EMI) environment by far-out better than digital computers that may encounter electronic latch up in their circuitry [3].

Note that: A latch-up is a type of short circuit which can occur in an Integrated Circuit (IC). More specifically it is the inadvertent creation of a low-impedance path between the power supply rails of a Metal–Oxide–Semiconductor Field-Effect Transistor (MOSFET) circuit, triggering a parasitic structure which disrupts proper functioning of the part, possibly even leading to its destruction due to over-current. A power cycle is required to correct this situation. This type of shortage in IC may also take place by and EMP events that is known as, a Single Event Latch-up, which is a latch-up caused by a single event upset, typically heavy ions or protons from cosmic rays or solar flares.

5.5.3. Nature of Algorithms in Computer

We are familiar with machines that perform the operations of arithmetic (like addition and multiplication), what we would call it, an arithmetical functions. It is possible to write a

computer program that performs arithmetical functions because these arithmetical functions (e.g., addition and multiplication) are computable functions. For a function to be computable, it must admit of an algorithm. An algorithm is a step-by-step mechanical process which, if followed faithfully, is guaranteed to produce the correct output (the "answer") for any input. The rules that a calculator follows when it performs arithmetic are all algorithms.

Consider multiplication. Any set of rules that takes two numbers as input and gives as output the product of those two numbers, will be a multiplication function. In this sense of function, we do not care how the rule produces the output; we only care what the output is for any given input. So, two different rules (two different algorithms) can both be multiplication functions. This means that there is more than one way to write a computer program for a calculator that can do multiplication. Here is one familiar set of rules -- one algorithm -- for performing multiplication (See Table 5-1):

Table 5-1. An Algorithm for doing Multiplication by Multiple Additions

STEP 1	LIST each input item in a column, aligning each input to the right, then draw a line beneath the last item.	4 4 +4
STEP 2	ADD the units digits. Place the unit digit of the sum beneath the line in the unit's digit column. If the number is greater than 9, place the 10's digit of the sum above the first number in 10's digit column. If there is no such column, place the 10's digit of the sum beneath the line in the 10's digit position. STOP.	4 4 +4 2
STEP 3	ADD the 10's digits. Place the 10's digit of the sum beneath the line in the 10's digit column. If the number is greater than 99, place the 100's digit of the sum above the first number in 100's digit column. If there is such column, place the 100's digit of the sum beneath the line in the 100's digit position. STOP.	

This is one set of rules for doing multiplication. However, there are others. You could also write a computer program to perform multiplication using the rules in Table 5-2.

Table 5-2. Another Algorithm for doing Multiplication that uses a "Look-up" Table

<table>
<tr><td>STEP 1</td><td>LIST each input item in a column, aligning each input to the right, then draw a line beneath the last item.</td><td>3
x 4</td></tr>
<tr><td>STEP 2</td><td>Consult multiplication table:
<table>
<tr><td>x</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr>
<tr><td>2</td><td>2</td><td>4</td><td>6</td><td>8</td><td>10</td></tr>
<tr><td>3</td><td>3</td><td>6</td><td>9</td><td>12</td><td>15</td></tr>
<tr><td>4</td><td>4</td><td>8</td><td>12</td><td>16</td><td>20</td></tr>
<tr><td>5</td><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td></tr>
</table></td><td>3
x 4</td></tr>
<tr><td>STEP 3</td><td>Locate the first number (3) in the horizontal column. Locate the second number (4) in the vertical column. Locate the number in the square that forms the intersection of the horizontal and vertical columns (12). Write that number below the line. STOP.</td><td></td></tr>
</table>

This shows that two different computer programs, following different rules (or algorithms), can both be giving exactly the same output for a given input. They perform the same general multiplication function, even though they do it different ways.

In summary, then, digital computers are machines that perform (or implement) algorithms. But there are other machines that may deserve the title, “computer,” even though they do not implement algorithms. We will call such computers “Non-classical” computers, and we turn to a discussion of these machines next [4].

5.5.4. Distributed (Not Local)

Physics sets certain limits on what is and is not computable. These limits are very far from having been, reached by current technologies. Whilst proposals for hyper-computation are almost certainly infeasible, there are, a number of non-classical approaches that do hold considerable promise. There is a range of possible architectures that could be implemented on silicon that are distinctly different from the von Neumann model. Beyond this, quantum simulators, which are the quantum equivalent of analogue computers, may be constructible in the near future. The models of physics used by many of the hyper-computing proposals (Beggs and Tucker (2007, 2006) [7, 8]; Bournez and Cosnard (1995)) [9] are classical rather than quantum.

Another characteristic of a non-classical computer is that the elements of the system that are carrying the information are not, localized in any particular place.

Note that: Distributed means spread or scattered out. With regard to computing, distributed means that key elements of the system are not grouped together and only loosely are connected.

5.5.5. Probabilistic (Not Deterministic)

Digital computers are designed to run algorithmic functions. Previously, we defined an algorithm as a step-by-step mechanical process which, if followed faithfully, is guaranteed to produce the correct result. Our red ball computer is of this kind. It is deterministic because it is so designed that it must always behave in exactly the same way. You get exactly the same output whenever your input is the same. Thus, if you have an effective counting algorithm (and the machinery is working properly) then you will always get the same result -- the laws of physics that control the operations of the equipment and the implementation of the software will guarantee that the output is entirely predictable.

This sort of computer does not always behave in exactly the same way, which means it is not *deterministic*.

In short:

A. A deterministic system is one in which the occurrence of all events is known with certainty. If the description of the system state at a particular point of time of its operation is given, the next state can be perfectly predicted.
B. A probabilistic system is one in which the occurrence of events cannot be perfectly predicted. Though the behavior of such a system can be described in terms of

probability, a certain degree of error is always attached to the prediction of the behavior of the system.

In a probabilistic system, unlike in a deterministic system, what has just occurred is not always an accurate predictor of what will transpire next. The weather, for example, is a probabilistic system, in that future events can only be imperfectly, predicted.

The theory of probabilistic systems by extension leads to probabilistic analysis and forecasting. A probabilistic system must be, analyzed according to the various possible outcomes and their relative probability of occurrence.

However, in mathematics and physics, a deterministic system is a system in which no randomness is involved in the development of future states of the system. A deterministic model will thus always produce the same output from a given starting condition or initial state. For example, Physical laws that are described by differential equations represent deterministic systems, even though the state of the system at a given point in time may be difficult to describe explicitly.

In quantum mechanics, the Schrödinger equation, which describes the continuous time evolution of a system's wave function, is deterministic. However, the relationship between a system's wave function and the observable properties of the system appears to be non-deterministic.

The systems studied in chaos theory are deterministic. If the initial state were known exactly, then the future state of such a system could theoretically be predicted. However, in practice, knowledge about the future state is limited by the precision with which the initial state can be measured, and chaotic systems are characterized by a strong dependence on the initial conditions [3]. This sensitivity to initial conditions can be measured with Lyapunov exponents.

Markov chains and other random walks are not deterministic systems, because their development depends on random choices.

A pseudorandom number generator is a deterministic algorithm, although its evolution is deliberately, made hard to predict. A hardware random number generator, however, may be non-deterministic.

In economics, the Ramsey–Cass–Koopmans model is deterministic. The stochastic equivalent is, known as Real Business Cycle theory.

5.5.6. Parallel (Not Serial)

Parallel computing is a type of computation in which many calculations or the executions of processes are, carried out simultaneously. Large problems can often be divided into smaller ones, which can then be solved at the same time. There are several different forms of parallel computing: bit-level, instruction-level, data, and task parallelism. Parallelism has been, employed for many years, mainly in high-performance computing, but interest in it has grown lately due to the physical constraints preventing frequency scaling. As power consumption (and consequently heat generation) by computers has become a concern in recent years, parallel computing has become the dominant paradigm in computer architecture, mainly in the form of multi-core processors.

Figure 5-18. IBM's Blue Gene/P Massively Parallel Supercomputer.

Parallel computing is closely related to concurrent computing—they are frequently used together, and often conflated, though the two are distinct: it is possible to have parallelism without concurrency (such as bit-level parallelism), and concurrency without parallelism (such as multitasking by time-sharing on a single-core CPU). In parallel computing, a computational task is typically broken down in several, often many, very similar subtasks that can be processed independently and whose results are combined afterwards, upon completion. In contrast, in concurrent computing, the various processes often do not address related tasks; when they do, as is typical in distributed computing, the separate tasks may have a varied nature and often require some inter-process communication during execution.

5.5.7. Summary

A further property that distinguishes many non-classical computers from the more familiar digital computers is the order in which processing is done. The standard paradigm for digital computers is that the operations to be, performed by the computer program must be done in one-step at a time and in a specific, regimented order.

You have just been, introduced to two basic types of computer: Digital computers and non-classical computers. As you further explore this curriculum, you will discover that both kinds of computers are used in a variety of ways to help shed light on the nature of human (and non-human) minds. One very important question that you will confront is this: Is the human brain a computer and, if so, what kind of computer is it? There are many, questions about computers that lie ahead.

5.6. Cognitive Computing

Cognitive computing is the simulation of human thought processes in a computerized model. Cognitive computing involves self-learning systems that use data mining, pattern

recognition and natural language processing to mimic the way the human brain works. The goal of cognitive computing is to create automated IT systems that are capable of solving problems without requiring human assistance.

Cognitive computing systems use machine-learning algorithms. Such systems continually acquire knowledge from the data fed into them by mining data for information. The systems refine the way they look for patterns and as well as the way they process data so they become capable of anticipating new problems and modeling possible solutions.

Cognitive computing is, used in numerous artificial intelligence (AI) applications, including expert systems, natural language programming, neural networks, robotics and virtual reality. The term cognitive computing is closely associated with IBM's cognitive computer system, Watson.

5.7. Artificial Intelligence and Brain Mechanisms

Artificial neural networks are parallel computational models comprised of densely interconnected adaptive processing units. These networks are fine-grained parallel implementation of nonlinear static or dynamic systems.

It is argued that the concept of intelligence can best be analyzed in terms of problem-solving behavior, which on the linguistic level is inference-making. An examination of the role of methodological advice in inference-making. The brain is considered as a language machine which contains mechanisms that implement the methodology of inference-making and operates on depressions in "cortical" language, so as to derive conclusions about what to expect and how to respond. An artificial neuron-device is then analyzed, and it is suggested how nets of such devices can be interpreted as mechanisms that form hypotheses, make predictions, and incorporate methods for making and improving inferences. Finally, some ideas are, presented on how to formulate an actual theory of artificial intelligence. A very important feature of these networks is their adaptive nature, where "*learning by example*" replaces "*programming*" in solving problems. This feature makes such computational models very appealing in application domains where one has little incomplete understanding of the problem to be solved but where training data is readily available. See Chapter 4 for different data warehousing and data mining.

Another key feature is the intrinsic parallel architecture that allows for fast computation of solutions when these networks are, implemented on parallel digital computers. or, ultimately, when implemented in customized hardware.

Artificial neural networks are viable computational models for a wide variety of problems. These include pattern classification, speech synthesis and recognitions, while they can be, used as part of resilience system to process data in real time to prevent any adversary impact on friendly environments of enterprises as well organizations, adaptive interfaces between humans and complex physical system [10-11].

Chief Information Officers (CIOs) play a dual role as the technology leader to enable business automation and innovation and as the IT functional leader. It is important for CIOs to understand the breadth and depth of this new technology landscape, with a directional sense of where it is going and a roadmap to take advantage of it for benifit of their organizations to be more resilience and robust.

In this book, we seek to define digital labor, describe its various forms and potential use cases, and provide pragmatic recommendations for CIOs who want to get started deploying digital labor within their companies

Furthermore, the application of these types of Artificial Neural Networks (ANN) can be seen in function approximation, image compression, associative memory, clustering, forecasting and prediction to prevent unfriendly events, combinational optimization, nonlinear system modeling, and control.

These networks are “neural” in the sense that they may have been inspired by neuroscience but not necessarily because they are faithful models of biologic neural or cognitive phenomena.

As Andrew Friedman, describes under a published paper with neuroscience special report and title of “The Fundamental Distinction between Brains and Turning Machines,” as follows.

The exponential growth in computing power in the past few decades has let to genuine reality that computers has surpassed human intelligence in many realms. This fact, along with other observations such as the apparent similarity between binary arithmetic and the behavior of neurons, and the digital nature of DNA, has led to the fairly widespread belief that the brain is just an organic computer or we may call it a wet computer, even though a highly complex one. See Figure 5-19.

What is DNA

Deoxyribonucleic Acid (DNA), a self-replicating material present in nearly all living organisms as the main constituent of chromosomes. It is the carrier of genetic information. The fundamental and distinctive characteristics or qualities of someone or something, especially when regarded as unchangeable. However, deoxyribonucleic acid is a molecule that carries the genetic instructions used in the growth, development, functioning and reproduction of all known living organisms and many viruses.

5-19. Robot of Future.

The logical extension of the idea of the brain as a computer extension of the idea of the brain as a computer is that one day, given sufficient complexity, computers will not only surpass us as human in computational ability, but will also achieve the status of conscious machines has become ingrained in the public consciousness through popular depictions of androids and other conscious robots in literature and film as it can be seen in Figure 5-20 as

artistic image. And indeed it is a belief that is taken seriously by many in the artificial intelligence community.

Figure 5-20. Artistic Image of Smart Robot.

The availability of powerful but cheap processing power on demand, coupled with advances in artificial intelligence, natural language processing, and exponential growth of data has created an opportunity to deploy digital labor to substitute or augment human labor and open the door for step-change improvements in costs, quality and speed for businesses across all industry sectors.

The “artificial neuron” is the basic building block and processing unit (i.e., CPU in classical computers of today) of an artificial neural network. It is necessary to understand the computational capabilities of these units.

To examine whether the brain is just an organic or wet computer, it becomes necessary to start from the theoretical definition of a computer, the Turning Machine, first envisioned by Alan Turing, whose work in the early 1950's formed the foundation for the theory of modern day computer science [13].

From view point of mathematics, a Turing Machine is an abstract algorithmic manipulator of intrinsically meaningless symbols, usually the binary digits or bits, 0 and 1, as we stated before. The idealized Turing consists of an infinite strip of tape made of squares containing 1's or 0's, a reading head that reads the bit in the present square, a track where the reading head can move between squares in both directions, and a writing instrument that can change the bit in a square from 1 to 0 or vice versa.

Algorithms, or programs, are enacted by some sequence of these basic operations, of which there turn out to be only 7 moves as follows:

1. Read 1,
2. Read 0,
3. Write 1,
4. Write 0,
5. Move left to position i
6. Move right to position j, and
7. Stop

Despite the high level of abstraction involved in this concept, all conventional computer software and indeed all standard home computers, from Macs to PCs, can be described in this framework as algorithms implemented on a Turing Machine.

Given this notion of a Turing Machine, the focus of this chapter then becomes, "*Are our brains merely Turing Machines*, *where our conscious minds are simply algorithmic program*?." However Freedman [13] argues that, in fact, our brains are fundamentally different than Turing Machines, and that a Turing Machine itself could never be conscious in principal. Never the less, this does not rule out the possibility of consciousness involving hardware other than neurons, or even the possibility of consciousness that we helped to create, only that consciousness itself can never be implemented on a Turing Machine alone. Next section possibly help us to begin a useful discussion how one might test in practice whether existing or possible future Turing Machine are conscious.

5.8. THINKING IN SILICON

Microchips that are designed, modeled on the brain and someday going to be manufactured at full scale, may excel at tasks that baffle today's computers. Computers that can efficiently process real-world data as images or speech could accelerate the progress of many artificial intelligence projects, including autonomous robots and mobile devices that like smart assistance are in horizon and becoming a necessity as part of our life. Figure 5-21 is a typical computer chip, made by IBM in 2011, features components that serve as 256 neurons and 262,144 synapses

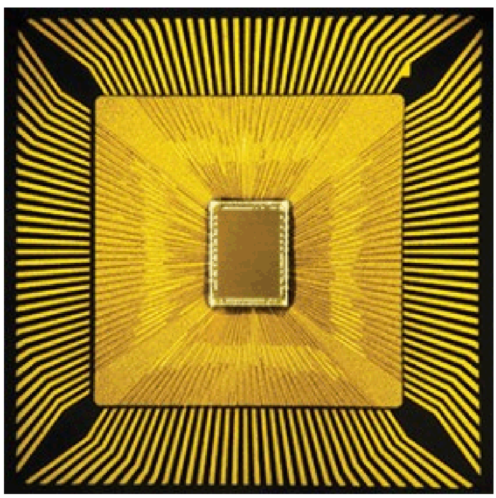


Figure 5-21. Typical Integrated Circuits.

Computers have emerged from back rooms and laboratories to help with writing, calculating, and play in homes and offices. These machines do simple, repetitive tasks, but machines still in the laboratory do much more. Artificial intelligence researchers say that computers can be made smart, and fewer and fewer people disagree. To understand our future, we must see whether artificial intelligence is as impossible as flying to the Moon.

Computers that can efficiently process real-world data such as images or speech could accelerate the progress of many artificial-intelligence projects, including autonomous robots and mobile devices that act like smart assistants. These are the new breed of computer generations with implementation of neural networking processors in them. This tells us that microchips modeled on the brain may excel at tasks that baffle today's computers.

The world stands on the threshold of a second computer age. New technology now moving out of the laboratory is starting to change the computer from a fantastically fast calculating machine to a device that mimics human thought processes - giving machines the capability to reason, make judgments, and even learn. Already this "artificial intelligence" is performing tasks once thought to require human intelligence.

Thinking machines need not resemble human beings in shape, purpose, or mental skills. Indeed, some artificial intelligence systems will show few traits of the intelligent liberal arts graduate, but will instead serve only as powerful engines of design. Nonetheless, understanding how human minds evolved from mindless matter will shed light on how machines can be made to think. Minds, like other forms of order, evolved through variation and selection. Figure 5-22 illustrates the history of computing and computers.

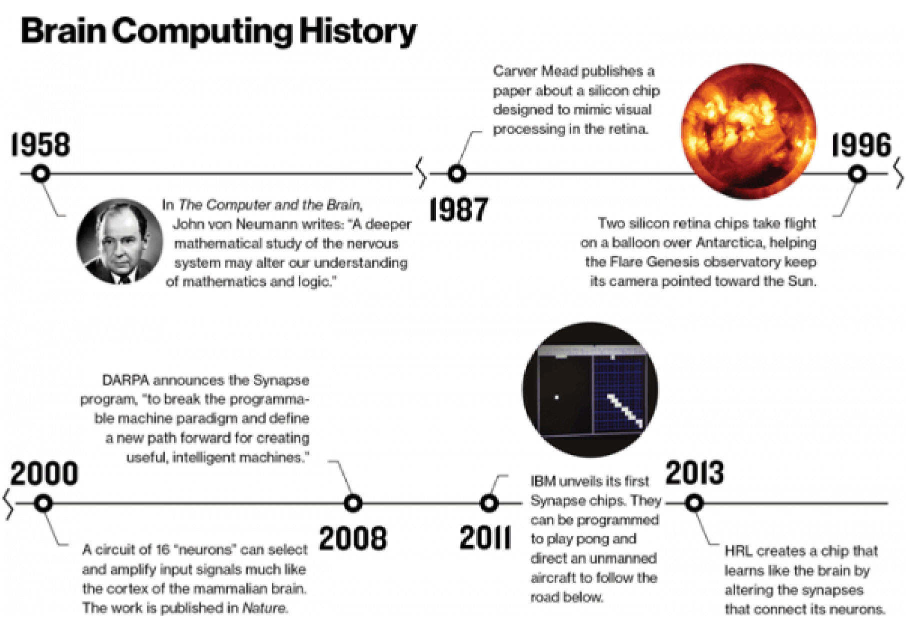


Figure 5-22. Brain Computing History.

An article published by Tom Simonite in December 16, 2013, under the title of *Thinking in Silicon* in journal of MIT Technology Review, presents the following opinions.

"Picture a person reading these words on a laptop in a coffee shop. The machine made of metal, plastic, and silicon consumes about 50 watts of power as it translates bits of information—a long string of 1s and 0s—into a pattern of dots on a screen. Meanwhile, inside that person's skull, a gooey clump of proteins, salt, and water uses a fraction of that power not only to recognize those patterns as letters, words, and sentences but to recognize the song playing on the radio.

Computers are incredibly inefficient at lots of tasks that are easy for even the simplest brains, such as recognizing images and navigating in unfamiliar spaces. Machines found in research labs or vast data centers can perform such tasks, but they are huge and energy-hungry, and they need specialized programming. Google recently made headlines with software that can reliably recognize cats and human faces in video clips, but this achievement required no fewer than 16,000 powerful processors.

A new breed of computer chips that operate more like the brain may be about to narrow the gulf between artificial and natural computation—between circuits that crunch through logical operations at blistering speed and a mechanism honed by evolution to process and act on sensory input from the real world. Advances in neuroscience and chip technology have made it practical to build devices that, on a small scale at least, process data the way a mammalian brain does. These "neuromorphic" chips may be the missing piece of many promising but unfinished projects in artificial intelligence, such as cars that drive themselves reliably in all conditions, and smartphones that act as competent conversational assistants.

"Modern computers are inherited from calculators, good for crunching numbers," says Dharmendra Modha, a senior researcher at IBM Research in Almaden, California. "Brains evolved in the real world." Modha leads one of two groups that have built computer chips with a basic architecture copied from the mammalian brain under a $100 million project called Synapse, funded by the Pentagon's Defense Advanced Research Projects Agency (DARPA).

The prototypes have already shown early sparks of intelligence, processing images very efficiently and gaining new skills in a way that resembles biological learning. IBM has created tools to let software engineers program these brain-inspired chips; the other prototype, at HRL Laboratories in Malibu, California, will soon be installed inside a tiny robotic aircraft, from which it will learn to recognize its surroundings.

The evolution of brain-inspired chips began in the early 1980s with Carver Mead, a professor at the California Institute of Technology and one of the fathers of modern computing. Mead had made his name by helping to develop a way of designing computer chips called very large scale integration, or VLSI, which enabled manufacturers to create much more complex microprocessors. This triggered explosive growth in computation power: computers looked set to become mainstream, even ubiquitous. But the industry seemed happy to build them around one blueprint, dating from 1945. The von Neumann architecture, named after the Hungarian-born mathematician John von Neumann, is designed to execute linear sequences of instructions. All today's computers, from smartphones to supercomputers, have just two main components: a central processing unit, or CPU, to manipulate data, and a block of random access memory, or RAM, to store the data and the instructions on how to manipulate it. The CPU begins by fetching its first instruction from memory, followed by the data needed to execute it; after the instruction is performed, the result is sent back to memory and the cycle repeats. Even multi-core chips that handle data in parallel are limited to just a few simultaneous linear processes.

That approach developed naturally from theoretical math and logic, where problems are solved with linear chains of reasoning. Yet it was unsuitable for processing and learning from large amounts of data, especially sensory input such as images or sound. It also came with built-in limitations: to make computers more powerful, the industry had tasked itself with building increasingly complex chips capable of carrying out sequential operations faster and faster, but this put engineers on course for major efficiency and cooling problems, because

speedier chips produce more waste heat. Mead, now 79 and a professor emeritus, sensed even then that there could be a better way. “The more I thought about it, the more it felt awkward,” he says, sitting in the office he retains at Caltech. He began dreaming of chips that processed many instructions—perhaps millions—in parallel. Such a chip could accomplish new tasks, efficiently handling large quantities of unstructured information such as video or sound. It could be more compact and use power more efficiently, even if it were more specialized for particular kinds of tasks. Evidence that this was possible could be found flying, scampering, and walking all around. “The only examples we had of a massively parallel thing were in the brains of animals,” says Mead.

Brains compute in parallel as the electrically active cells inside them, called neurons, operate simultaneously and unceasingly. Bound into intricate networks by threadlike appendages, neurons influence one another’s electrical pulses via connections called synapses. When information flows through a brain, it processes data as a fusillade of spikes that spread through its neurons and synapses. You recognize the words in this paragraph, for example, thanks to a particular pattern of electrical activity in your brain triggered by input from your eyes. Crucially, neural hardware is also flexible: new input can cause synapses to adjust so as to give some neurons more or less influence over others, a process that underpins learning. In computing terms, it’s a massively parallel system that can reprogram itself.

Ironically, though he inspired the conventional designs that endure today, von Neumann had also sensed the potential of brain-inspired computing. In the unfinished book The Computer and the Brain, published a year after his death in 1957, he marveled at the size, efficiency, and power of brains compared with computers. “Deeper mathematical study of the nervous system … may alter the way we look on mathematics and logic,” he argued. When Mead came to the same realization more than two decades later, he found that no one had tried making a computer inspired by the brain. “Nobody at that time was thinking, ‘How do I build one?’” says Mead. “We had no clue how it worked.”

Mead finally built his first neuromorphic chips, as he christened his brain-inspired devices, in the mid-1980s, after collaborating with neuroscientists to study how neurons process data. By operating ordinary transistors at unusually low voltages, he could arrange them into feedback networks that looked very different from collections of neurons but functioned in a similar way. He used that trick to emulate the data-processing circuits in the retina and cochlea, building chips that performed tricks like detecting the edges of objects and features in an audio signal. But the chips were difficult to work with, and the effort was limited by chip-making technology. With neuromorphic computing still just a curiosity, Mead moved on to other projects. “It was harder than I thought going in,” he reflects. “A fly’s brain doesn’t look that complicated, but it does stuff that we to this day can’t do. That’s telling you something.”

5.9. Introduction to Biochip

A biochip is a collection of miniaturized test sites (microarrays) arranged on a solid substrate that permits many tests to be performed at the same time in order to achieve higher throughput and speed. Typically, a biochip's surface area is no larger than a fingernail. Like a

computer chip that can perform millions of mathematical operations in one second, a biochip can perform thousands of biological reactions, such as decoding genes, in a few seconds.

In molecular biology, biochips are essentially miniaturized laboratories that can perform hundreds or thousands of simultaneous biochemical reactions. Biochips enable researchers to quickly screen large numbers of biological analyses for a variety of purposes, from disease diagnosis to detection of bioterrorism agents. Digital microfluidic biochips [13] have become one of the most promising technologies in many biomedical fields. In a digital microfluidic biochip, a group of (adjacent) cells in the microfluidic array can be configured to work as storage, functional operations, as well as for transporting fluid droplets dynamically.

Note that, Microfluidic biochips offer a promising platform for massively parallel DNA analysis, automated drug discovery, and real-time bimolecular recognition. Current techniques for full-custom design of droplet-based "digital" biochips do not scale well for concurrent assays and for next-generation, System-On-Chip (SOC) designs that are expected to include microfluidic components [13].

The advantages of scalability and reconfigurability make digital microfluidic biochips a promising platform for massively parallel DNA analysis, automated drug discovery, and real-time bimolecular detection. As the use of digital microfluidic biochips increases, their complexity is expected to become significant due to the need for multiple and concurrent assays on the chip, as well as more sophisticated control for resource management.

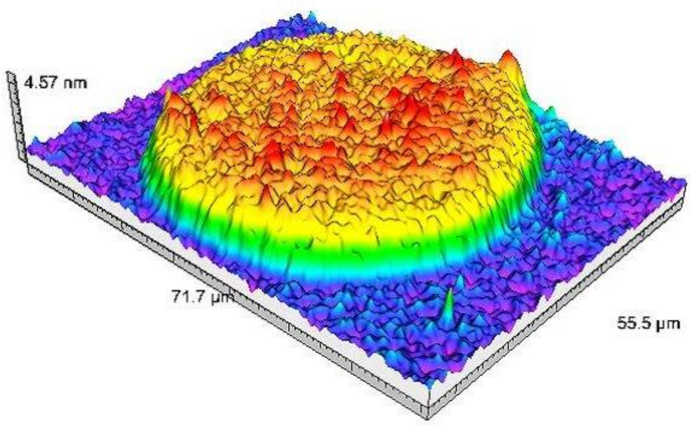


Figure 5-23. 3D Surface Image of a DNA Biochip.

Figure 5-23 is illustration of a typical 3D surface of Microarray Fabrication of DNA Biochip. The dense, two-dimensional grid of biosensors — is the critical component of a biochip platform. Typically, the sensors are deposited on a flat substrate, which may either be passive (e.g., silicon or glass) or active, the latter consisting of integrated electronics or micromechanical devices that perform or assist signal transduction. Surface chemistry is used to covalently bind the sensor molecules to the substrate medium. The fabrication of microarrays is non-trivial and is a major economic and technological hurdle that may ultimately decide the success of future biochip platforms. The primary manufacturing challenge is the process of placing each sensor at a specific position (typically on a Cartesian grid) on the substrate. Various means exist to achieve the placement, but typically robotic micro-pipetting [14] or micro-printing [15] systems are used to place tiny spots of sensor

material on the chip surface. Because each sensor is unique, only a few spots can be placed at a time. The low-throughput nature of this process results in high manufacturing costs.

Certainly the fabrication of these chips requires advanced semiconductor fabrication utilizing process such as Physical Vapor Deposition (PVD), Chemical Vapor Deposition (CVD) and Chemical Mechanical Polishing (CMP) to push the real states of these chips to way below sub-micron technology, and avoid any void between different layers of the masks printed into silicon.

A genetic biochip is designed to "freeze" into place the structures of many short strands of DNA (deoxyribonucleic acid), the basic chemical instruction that determines the characteristics of an organism. Effectively, it is used as a kind of "test tube" for real chemical samples. A specially designed microscope can determine where the sample hybridized with DNA strands in the biochip. Biochips helped to dramatically accelerate the identification of the estimated 80,000 genes in human DNA, an ongoing world-wide research collaboration known as the Human Genome Project. The microchip is described as a sort of "word search" function that can quickly sequence DNA.

In addition to genetic applications, the biochip is being used in toxicological, protein, and biochemical research. Biochips can also be used to rapidly detect chemical agents used in biological warfare so that defensive measures can be taken.

5.9.1. Neurons Inside

Scientists at IBM's facility in Almaden California lead a project under DARPA and are tasked to break the computing industry's von Neumann dependency. The basic approach is similar to Mead's: build silicon chips with elements that operate like neurons. But they have the benefit of advances in neuroscience and chip making. "Timing is everything; it was not quite right for Carver," says Modha, who is the leading scientist here at IBM.

IBM makes neuromorphic chips by using collections of 6,000 transistors to emulate the electrical spiking behavior of a neuron and then wiring those silicon neurons together. Modha's strategy for combining them to build a brain like system is inspired by studies on the cortex of the brain, the wrinkly outer layer. Although different parts of the cortex have different functions, such as controlling language or movement, they are all, made up of so-called micro-columns, repeating clumps of 100 to 250 neurons. Modha unveiled his version of a micro-column in 2011. A speck of silicon little bigger than a pinhead, it contained 256 silicon neurons and a block of memory that defines the properties of up to 262,000 synaptic connections between them. Programming those synapses correctly can create a network that processes and reacts to information much as the neurons of a real brain do.

Setting that chip to work on a problem involves programming a simulation of the chip on a conventional computer and then transferring the configuration to the real chip. In one experiment, the chip could recognize handwritten digits from 0 to 9, even predicting which number someone was starting to trace with a digital stylus. In another, the chip's network was programmed to play a version of the video game Pong. In a third, it directed a small-unmanned aerial vehicle to follow the double yellow line on the road approaching IBM's lab. None of these feats are beyond the reach of conventional software, but they were achieved using a fraction of the code, power, and hardware that would normally be required.

IBM researchers are testing early versions of a more complex chip, made from a grid of neurosynaptic cores tiled into a kind of rudimentary cortex—over a million neurons altogether. Last summer, IBM also announced a neuromorphic programming architecture based on modular blocks of code called corelets. The intention is for programmers to combine and tweak corelets from a pre-existing menu, to save them from wrestling with silicon synapses and neurons. Over 150 corelets have already been, designed, for tasks ranging from recognizing people in videos to distinguishing the music of Beethoven and Bach.

5.9.2. Neurosynaptic Chip or Cognitive Chip

A neurosynaptic chip, also known as a cognitive chip, is a computer processor that functions more like a biological brain than a typical CPU does. Unlike cognitive computing, which is made to emulate the thought and learning of humans through software, neurosynaptic chips are made to function like human brains on the hardware level.

While a typical computer works well for language, mathematical and data analytics processing, it can take a lot of work for it to perform tasks that even simple biological brains are efficient at. A neurosynaptic chip is more efficient at these tasks, which include pattern recognition and sensory processing and learning.

Here's how IBM describes the new architecture:

> "IBM's brain-inspired architecture consists of a network of neurosynaptic cores. Cores are distributed and operate in *parallel*. Cores operate—without a clock—in an event-driven fashion. Cores integrate memory, computation, and communication. Individual cores can fail and yet, like the brain, the architecture can still function. Cores on the same chip communicate with one another via an on-chip event-driven network. Chips communicate via an inter-chip interface leading to seamless availability like the cortex, enabling creation of scalable neuromorphic systems."

Especially in mobile technology, where processing and power are limited, the neurosynaptic chip stands to revolutionize abilities. Tasks like selecting the best produce or finding repair points in electronics could be carried out using Smartphone cameras. For traditional computer architectures, it takes a supercomputer to perform these tasks, consuming massive amounts of power in the process. Neurosynaptic chips make this possible with a tenth of the energy requirements.

The new type of chip also has promise for supercomputing applications. IBM has a neurosynaptic processor project, called Brainpower. MIT has simulated a functioning brain synapse in their quest for truly intelligent systems. A goal of the IBM project is a trillion synapses with only 4kW.

Currently in its 2nd generation, the neurosynaptic chip has made impressive strides in its specs:

- **Generation 1 -** 256 programmable neurons, 262114 programmable synapses, 1 neurosynaptic core.
- **Generation 2** – 1 million programmable neurons, 256 million programmable synapses, 4096 neurosynaptic cores.

5.9.3. Learning Machines

HRL is another company in Southern California, located in Santa Monica, that has a DARPA's project that is aimed to make chips, which are suppose to mimic human brain even more closely. This company was founded by Hughes Aircraft and now operates as joint venture of General Motors and Boeing.

The HRL chip mimics two learning phenomena in brains. One is that neurons become more or less sensitive to signals from another neuron depending on how frequently those signals arrive. The other is more complex: a process believed to support learning and memory, known as spike-timing-dependent plasticity. This causes neurons to become more responsive to other neurons that have tended to closely, match their own signaling activity in the past. If groups of neurons are working together constructively, the connections between them strengthen, while less useful connections fall dormant.

Results from experiments with simulated versions of the chip are impressive. The chip played a virtual game of Pong, just as IBM’s chip did. But unlike IBM’s chip, HRL’s wasn’t programmed to play the game—only to move its paddle, sense the ball, and receive feedback that either rewarded a successful shot or punished a miss. A system of 120 neurons started out flailing, but within about five rounds it had become a skilled player. In this case “You do not program it.” “You just say ‘Good job,’ ‘Bad job,’ and it figures out what it should be doing.” If extra balls, paddles, or opponents are added, the network quickly adapts the changes.

This approach might eventually let engineers create a robot that stumbles through a kind of “childhood,” figuring out how to move around and navigate. “You cannot capture the richness of all the things that happen in the real-world environment, so you should make the system deal with it directly,” as the inventor scientist of this chip declare. Identical machines could then incorporate whatever the original one has learned. However, leaving robots some ability to learn after that point could also be useful. That way they could adapt if damaged, or adjust their gait to different kinds of terrain.

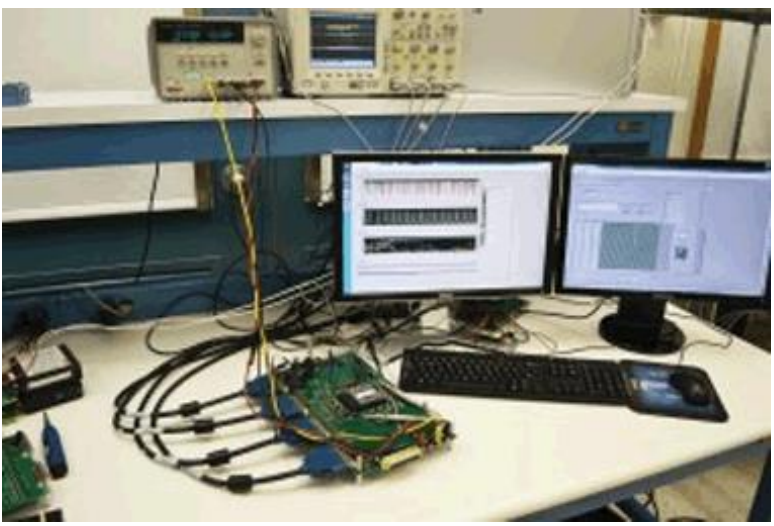


Figure 5-24. HRL Artificial Neurons Chip.

Figure 5-24, is illustration of bench mark of HRL chip sits in tangle of wires and the activity of its 576 artificial neurons appears on a computer screen as a parade of spikes, and Electro Encephalo-Graphy (EEG) for a silicon brain. The HRL chip has neurons and synapses much like IBM's. But like the neurons in your own brain, those on HRL's chip adjust their synaptic connections -hen exposed to new data. In other words, the chip learns through experience.

The first real test of this vision for neuromorphic computing was tested in summer of 2014, when the HRL chip was scheduled to escape its lab bench and take flight in a palm-sized aircraft with flapping wings, called a *Snipe*. As a human remotely pilots the craft through a series of rooms, the chip will take in data from the craft's camera and other sensors. At some point, the chip will be given a signal that means "Pay attention here." The next time the Snipe visits that room, the chip should turn on a light to signal that it remembers. Performing this kind of recognition would normally require too much electrical and computing power for such a small craft.

5.9.4. Alien Intelligence

Despite the Synapse chips' modest but significant successes, it is still unclear whether scaling up these chips will produce machines with more sophisticated brain like faculties. And some critics doubt it will ever be possible for engineers to copy biology closely enough to capture these abilities.

Neuroscientist Henry Markram, who discovered spike-timing-dependent plasticity, has attacked Modha's work on networks of simulated neurons, saying their behavior is too simplistic. He believes that successfully emulating the brain's faculties requires copying synapses down to the molecular scale; the behavior of neurons is influenced by the interactions of dozens of ion channels and thousands of proteins, he notes, and there are numerous types of synapses, all of which behave in nonlinear, or chaotic, ways. In Markram's view, capturing the capabilities of a real brain would require scientists to incorporate all those features. Figure 5-25 is depiction of IBM simulation of long-range neural pathways in a macaque monkey to guide the design of neuromorphic chips.

Figure 5-25. IBM Simulation of Long-Range-Neural Pathways.

The DARPA teams counter that they do not have to capture the full complexity of brains to get useful things done, and that successive generations of their chips can be expected to come closer to representing biology. HRL hopes to improve its chips by enabling the silicon neurons to regulate their own firing rate as those in brains do, and IBM is wiring the connections between cores on its latest neuromorphic chip in a new way, using insights from simulations of the connections between different regions of the cortex of a macaque.

Modha who is researcher at IBM, believes these connections could be important to higher-level brain functioning. Yet even after such improvements, these chips will still be far from the messy, complex reality of brains. It seems unlikely that microchips will ever match brains in fitting 10 billion synaptic connections into a single square centimeter, even though HRL is experimenting with a denser form of memory based on exotic devices known as memristors. Furthermore today's' approach to neural type chip is that, *the traditional approach is to add more computational capability and stronger algorithms, but that no longer scales*.

They may be alien, but IBM’s head of research strategy, Zachary Lemnios, predicts that we’ll want to get familiar with them soon enough. Many large businesses already feel the need for a new kind of computational intelligence, he says: “The traditional approach is to add more computational capability and stronger algorithms, but that just doesn’t scale, and we’re seeing that.” As examples, he cites Apple’s Siri personal assistant and Google’s self–driving cars. These technologies are not very sophisticated in how they understand the world around them, Lemnios says; Google’s cars rely heavily on preloaded map data to navigate, while Siri (Apple IPhone) taps into distant cloud servers for voice recognition and language processing, causing noticeable delays.

Today the cutting edge of artificial–intelligence software is a discipline known as “*Deep Learning*,” embraced by Google and Facebook, among others. It involves using software to simulate networks of very basic neurons on normal computer architecture (see “10 Breakthrough Technologies: ***Deep Learning***,” May/June 2013). But that approach, which produced Google’s cat-spotting software, relies on vast clusters of computers to run the simulated neural networks and feed them data. Neuromorphic machines should allow such faculties to be packaged into compact, efficient devices for situations in which it’s impractical to connect to a distant data center. IBM is already talking with clients interested in using neuromorphic systems. Security video processing and financial fraud prediction are at the front of the line, as both require complex learning and real-time pattern recognition.

Whenever and however neuromorphic chips are finally used, it will most likely be in collaboration with von Neumann machines. Numbers will still need to be, crunched, and even in systems faced with problems such as analyzing images, it will be easier and more efficient to have a conventional computer in command. Neuromorphic chips could then be, used for particular tasks, just as a brain relies on different regions specialized to perform different jobs.

As has usually been the case throughout the history of computing, the first such systems will probably be, deployed in the service of the U.S. military. “It’s not mystical or magical,” Gill Pratt, who manages the Synapse project at DARPA, says of neuromorphic computing. “It’s an architectural difference that leads to a different trade-off between energy and performance.” Pratt says that UAVs, in particular, could use the approach. Neuromorphic chips could recognize landmarks or targets without the bulky data transfers and powerful conventional computers now needed to process imagery. “Rather than sending video of a

bunch of guys, it would say, 'There's a person in each of these positions—it looks like they're running,'" he says.

This vision of a new kind of computer chip is one that both Mead and von Neumann would surely recognize.

5.10. Implementing Human-Like Intuition Mechanism in Artificial Intelligence

One of the serious problems in machine learning is the ability to understand and interpret past knowledge for accurately solving current problems or predicting possible events. Current algorithms and models cannot obtain the results as good as human intuition does. Most of these models are logic driven and are time dependent. They lack the ability to give consistently accurate results because on one hand, when information is not sufficient for drawing any conclusion, logic process simply gets stuck. On the other hand, time is a crucial constraint for real life scenarios, and logic process is slow because it has a large search space and a lot of calculation steps. These constraints indicate a serious need for faster models to resolve such limitations in machine learning [16].

5.11. Machine Learning

One of the serious problems in machine learning is the ability to understand and interpret past knowledge for accurately solving current problems or predicting possible events. Current algorithms and models cannot obtain the results as good as human intuition does. Most of these models are logic driven and are time dependent. They lack the ability to give consistently accurate results because on one hand, when information is not sufficient for drawing any conclusion, logic process simply gets stuck. On the other hand, time is a crucial constraint for real life scenarios, and logic process is slow because it has a large search space and a lot of calculation steps. These constraints indicate a serious need for faster models to resolve such limitations in machine learning. Little work has been done in the study of the intuition-based methods in AI and machine learning. It is explained by researchers in this field, that the variations in statistical intuition and statistical knowledge [17].

Clearly, intuition as a process is prone to incorrect values, and the correctness depends on various factors, especially the mapping of the correct element of the past experiences (or combination of them). From what we have learned and has shown experimentally that people have a tendency of not thinking hard. They seem to be inclined toward accepting what comes first in their mined without any proper and logical thinking [17].

Common Sense is defined as the ability to perceive possible consequences in a short period of time from a wide range of possibilities [18]. This explanation assures us that the normal process of thinking is based on the same principles. However, this explanation does not throw any light on the concept of intuition with a mathematical or logical explanation. Sloman [19] argues that the idea of comparing intuition with the concept of simulation, perception using analogical representation, is prone to several loopholes.

These include issues in non-logical reasoning, which is a central issue in intuition. Sloman [19] further argues that philosophy cannot be related to AI for finding answers to such higher level intelligence e.g., intuition. Moreover, the concept of intuition as suggested by the above experts focus largely on the concept and not on the representation and use of entities in the process. The concept of unknown entities [20-21] needs to be handled properly by them in order to be able to consider intuition as an effective problem-solving strategy. One of these examples is IBSEAD, which considers the presence of unknown entities for problem solving. Such concept of valid inference has not been considered for unknown entities by Sloman [19] in his views on AI and Philosophy.

In summary, the dramatic growth in practical applications for machine learning over the last ten years has been accompanied by many important developments in the underlying algorithms and techniques. For example Bayesian methods have grown from a specialist niche to become mainstream, while graphical models have emerged as a general framework for describing and applying probabilistic techniques and off course the more powerful technique that recently has drawn a lot of attention in this matter is Fuzzy Logic (FL) of Type II which is under serious investigation and research by AI scientist and Artificial Neural Network (ANN) folks.

However, the practical applicability of Bayesian methods has been greatly enhanced by development of a range of approximate inference algorithms such as variational Bays and expectation, while new models based on kernels have had a significant impact on both algorithms and applications [22].

5.12. DEEP LEARNING

Deep learning is a new era for next generation of computers, with massive amounts of computational power, where machines can now recognized objects and translated speech in real-time. It is suffice to say that, the Artificial Intelligence is finally getting smart. The basic idea—that software can simulate the neocortex's large array of neurons in an artificial "neural network"—is decades old, and it has led to as many disappointments as breakthroughs. But because of improvements in mathematical formulas and increasingly powerful computers, computer scientists can now model many more layers of virtual neurons than ever before.

With this greater depth, they are producing remarkable advances in speech and image recognition. Last June, a Google deep-learning system that had been shown 10 million images from YouTube videos proved almost twice as good as any previous image recognition effort at identifying objects such as cats. Google also used the technology to cut the error rate on speech recognition in its latest Android mobile software. In October, Microsoft chief research officer Rick Rashid wowed attendees at a lecture in China with a demonstration of speech software that transcribed his spoken words into English text with an error rate of 7 percent, translated them into Chinese-language text, and then simulated his own voice uttering them in Mandarin. That same month, a team of three graduate students and two professors won a contest held by Merck to identify molecules that could lead to new drugs. The group used deep learning to zero in on the molecules most likely to bind to their targets.

Figure 5-26. Robots Behaving as Human.

Deep-learning software attempts to mimic the activity in layers of neurons in the neocortex, the wrinkly 80 percent of the brain where thinking occurs. The software learns, in a very real sense, to recognize patterns in digital representations of sounds, images, and other data. Google in particular has become a magnet for deep learning and related AI talent, where they are taking the idea of deep learning to image recognition, search, and natural-language understanding.

All this has normally cautious AI researchers hopeful that intelligent machines may finally escape the pages of science fiction. Indeed, machine intelligence is starting to transform everything from communications and computing to medicine, manufacturing, and transportation. The possibilities are apparent in IBM's Jeopardy!-winning Watson computer, which uses some deep-learning techniques and is now being trained to help doctors make better decisions. Microsoft has deployed deep learning in its Windows Phone and Bing voice search.

Extending deep learning into applications beyond speech and image recognition will require more conceptual and software breakthroughs, not to mention many more advances in processing power. And we probably won't see machines we all agree can think for themselves for years, perhaps decades—if ever. But for now, says Peter Lee, head of Microsoft Research USA, "deep learning has reignited some of the grand challenges in artificial intelligence."

5.13. BUILDING A BRAIN

There have been many competing approaches to those challenges. One has been to feed computers with information and rules about the world, which required programmers to laboriously write software that is familiar with the attributes of, say, an edge or a sound. That

took lots of time and still left the systems unable to deal with ambiguous data; they were limited to narrow, controlled applications such as phone menu systems that ask you to make queries by saying specific words.

Neural networks, developed in the 1950s not long after the dawn of AI research, looked promising because they attempted to simulate the way the brain worked, though in greatly simplified form. A program maps out a set of virtual neurons and then assigns random numerical values, or "weights," to connections between them. These weights determine how each simulated neuron responds—with a mathematical output between 0 and 1—to a digitized feature such as an edge or a shade of blue in an image, or a particular energy level at one frequency in a phoneme, the individual unit of sound in spoken syllables.

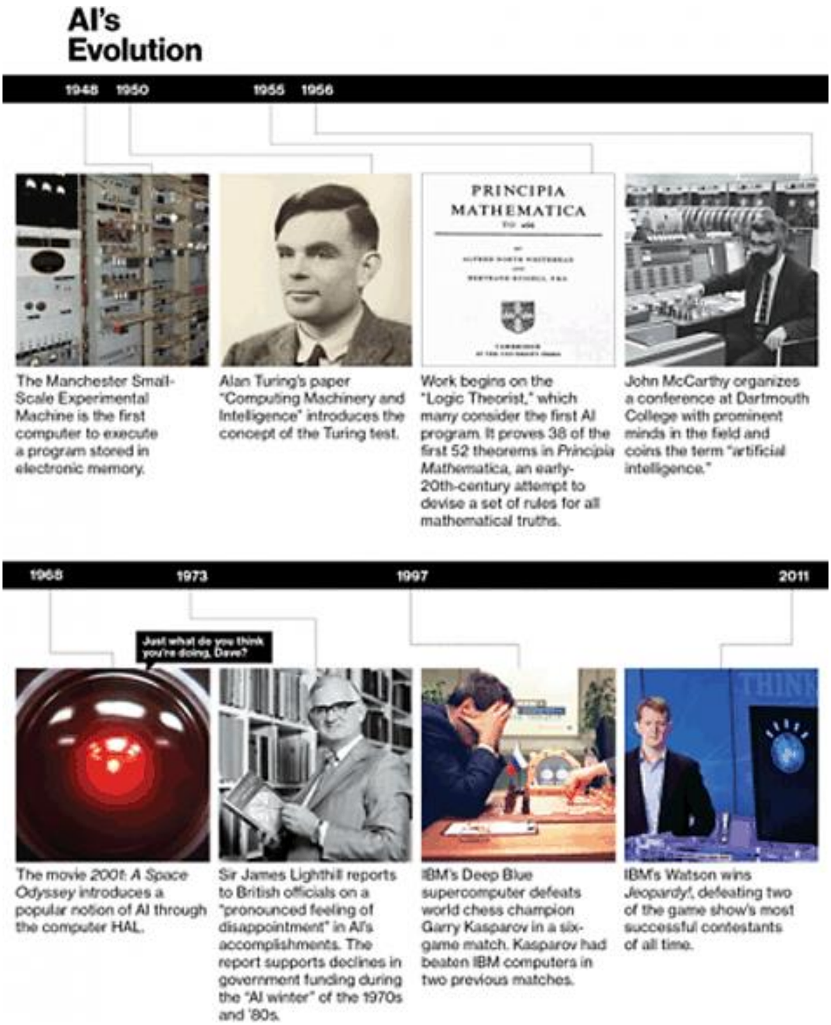


Figure 5-27. Artificial Intelligences Evolutions.

Programmers would train a neural network to detect an object or phoneme by blitzing the network with digitized versions of images containing those objects or sound waves containing those phonemes. If the network didn't accurately recognize a particular pattern, an algorithm would adjust the weights. The eventual goal of this training was to get the network to

consistently recognize the patterns in speech or sets of images that we humans know as, say, the phoneme “d” or the image of a dog. This is much the same way a child learns what a dog is by noticing the details of head shape, behavior, and the like in furry, barking animals that other people call dogs.

But early neural networks could simulate only a very limited number of neurons at once, so they could not recognize patterns of great complexity. They languished through the 1970s.

In the mid-1980s, Hinton and others helped spark a revival of interest in neural networks with so-called “deep” models that made better use of many layers of software neurons. But the technique still required heavy human involvement: programmers had to label data before feeding it to the network. And complex speech or image recognition required more computer power than was then available.

Finally, however, in the last decade -Hinton and other researchers made some fundamental conceptual breakthroughs. In 2006, Hinton developed a more efficient way to teach individual layers of neurons. The first layer learns primitive features, like an edge in an image or the tiniest unit of speech sound. It does this by finding combinations of digitized pixels or sound waves that occur more often than they should by chance. Once that layer accurately recognizes those features, they’re fed to the next layer, which trains itself to recognize more complex features, like a corner or a combination of speech sounds. The process is repeated in successive layers until the system can reliably recognize phonemes or objects.

Like cats. Last June, Google demonstrated one of the largest neural networks yet, with more than a billion connections. A team led by Stanford computer science professor Andrew Ng and Google Fellow Jeff Dean showed the system images from 10 million randomly selected YouTube videos. One simulated neuron in the software model fixated on images of cats. Others focused on human faces, yellow flowers, and other objects. And thanks to the power of deep learning, the system identified these discrete objects even though no humans had ever defined or labeled them.

What stunned some AI experts, though, was the magnitude of improvement in image recognition. The system correctly categorized objects and themes in the -YouTube images 16 percent of the time. That might not sound impressive, but it was 70 percent better than previous methods. And, Dean notes, there were 22,000 categories to choose from; correctly slotting objects into some of them required, for example, distinguishing between two similar varieties of skate fish. That would have been challenging even for most humans. When the system was asked to sort the images into 1,000 more general categories, the accuracy rate jumped above 50 percent.

5.14. The Rise of Artificial Intelligence and Its Threat to Humanity

Stephen Hawking, Britain’s pre-eminent theoretical physicist told the BBC in 2014, “The development of full artificial intelligence could spell the end of the human race. It would take off on its own, and re-design itself at an ever-increasing rate. Humans, who are limited by slow biological evolution, couldn’t compete, and would be superseded” [23].

But before you start building a bunker and preparing yourself for the imminent robot apocalypse, you should know that today's Artificial Intelligence (AI) technology is a long way from the ominous prophecy Hawking warned us about. Current AI technology depends heavily on human assistance to be successful, and serves largely to make humans more effective and productive [24].

Figure 5-28. Artistic Image of Human Alike Artificial Intelligence Robot.

Today's artificial intelligence with their new approaches of artificial neural networking promising a new generation of robots that by far smarter than the yesterday ones. There is a big push by multi-major corporation such as Google, Apple and recently Uber to go toward autonomous vehicle, which you are able to observer them on street of Northern Bay Area in California and Phoenix Arizona. You can also see the automated trading, and automated manufacturing of cars factories and other manufactures that have implemented robots in their assembly lines.

But Artificial Intelligence (AI) promises much more – including being man's best friend. BigDog was a robot developed in 2008, funded by the US Defense Advanced Research Projects Agency and the US Army Research Laboratory's RCTA program. It was designed to walk and climb – skills humans master instinctively at an early age, but which cannot easily be programmed into a machine. Instead, researchers applied AI techniques to enable it to "learn." Imagine a computer that can think better than humans; that can make profound cognitive decisions at lightning speed. Such a machine could better serve mankind. But would it? "AI that can run 1,000 times faster than humans can earn 1,000 times more than people," according to Stuart Armstrong, research fellow at the Future of Humanity Institute. "It can make 100 copies of itself." This ability to think fast and make copies of itself is a potent

combination – and one than could have a profound effect on humanity. Figure 5-29 is illustration of Army BigDog

Figure 5-29. Image of BigDog from the DARPA Strategic Plan (2007).

BigDog is a dynamically stable quadruped robot created in 2005 by Boston Dynamics with Foster-Miller, the NASA Jet Propulsion Laboratory, and the Harvard University Concord Field Station. It was funded by DARPA, but the project was shelved after the BigDog was deemed too loud for combat. BigDog, is the most advanced rough-terrain robot on the Earth so far and uses artificial intelligence to learn how to walk on a number of difficult terrains – such as on ice.

Now we can ask from business and total cost of ownership, would it makes that supper intelligent CIOs could mark the end of the HR department? AI will eventually be able to predict any move we make. Do the CIOs to start thinking about business impact of smart machines that exhibit AI behavior?. Advanced capabilities afforded by artificial intelligence will enhance today's smart devices to display goal-seeking and self-learning behavior, rather than a simple sense and respond. For CIOs, we regard autonomous business as a logical extension of current automated processes and to be compliment to their proposed business resilience system (BRS) by these authors (Zohuri and Moghaddam) [10] to increase efficiency to prevent any advisory action against their organizations, rather than simply to replace a human workforce.

For most people, AI is slanted to what you see on screen. But from a business perspective, we are far away from this in reality. In fact, there is no reason why a super-intelligent AI machine could not act like a CEO or manager, directing humans to do tasks where creativity or manual dexterity is important.

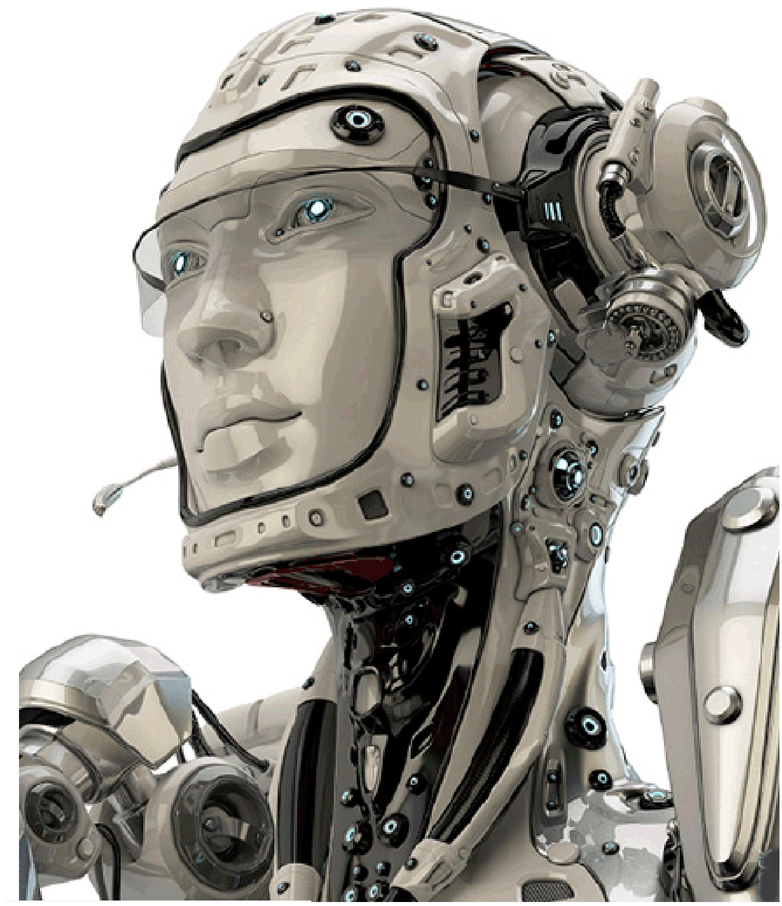


Figure 5-30. Super Intelligent Robot of Future.

This may sound like a plot from Channel 4 sci-fi drama Humans: "Even if the AI is nominally under human control, even if we can reprogram it or order it around, such theoretical powers will be useless in practice. This is because the AI will eventually be able to predict any move we make and could spend a lot of effort manipulating those who have 'control' over it." So back to man's best friend. We should not be afraid of the metal-clad robot with an Austrian accent that Arnold Schwarzenegger depicted in *The Terminator* move. For us, a super-intelligent machine taking the form of a dog and biting the proverbial hand that feeds it is a far more plausible way in which machines could eventually rule the world.

Although Cognitive Computing is a growing technology in today's market, is that a road for the robots of future to be smarter than their own creator is mainly, the human. As we stated in previously in this chapter, Cognitive computing is the simulation of human thought processes in a computerized model. Cognitive computing involves self-learning systems that use data mining, pattern recognition and natural language processing to mimic the way the

human brain works. The goal of cognitive computing is to create automated IT systems that are capable of solving problems without requiring human assistance.

Cognitive computing systems use machine learning algorithms. Such systems continually acquire knowledge from the data fed into them by mining data for information. The systems refine the way they look for patterns and as well as the way they process data so they become capable of anticipating new problems and modeling possible solutions.

Cognitive computing is used in numerous artificial intelligence (AI) applications, including expert systems, natural language programming, neural networks, robotics and virtual reality. The term cognitive computing is closely associated with IBM's cognitive computer system, Watson.

Watson is an IBM supercomputer that combines artificial intelligence (AI) and sophisticated analytical software for optimal performance as a "question answering" machine. The supercomputer is named for IBM's founder, Thomas J. Watson.

The Watson supercomputer processes at a rate of 80 teraflops (trillion floating-point operations per second). To replicate (or surpass) a high-functioning human's ability to answer questions, Watson accesses 90 servers with a combined data store of over 200 million pages of information, which it processes against six million logic rules. The device and its data are self-contained in a space that could accommodate 10 refrigerators.

Watson's key components include:

- Apache UIMA (Unstructured Information Management Architecture) frameworks, infrastructure and other elements required for the analysis of unstructured data
- Apache's Hadoop, a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment
- SUSE Enterprise Linux Server 11, the fastest available Power7 processor operating system
- 2,880 processor cores.
- 15 terabytes of RAM.
- 500 gigabytes of preprocessed information
- IBM's DeepQA software, which is designed for information retrieval that incorporates natural language processing and machine learning

Applications for the Watson's underlying cognitive computing technology are almost endless. Because the device can perform text mining and complex analytics on huge volumes of unstructured data, it can support a search engine or an expert system with capabilities far superior to any previously existing. In May 2016, Baker Hostetler, a century-old Ohio-based law firm, signed a contract for a legal expert system based on Watson to work with its 50-human bankruptcy team. ROSS can mine data from about a billion text documents, analyze the information and provide precise responses to complicated questions in less than three seconds. Natural language processing allows the system to translate legalese to respond to the lawyers' questions. ROSS' creators are adding more legal modules; similar expert systems are transforming medical research.

To showcase its abilities, Watson challenged two top-ranked players on Jeopardy! and beat champions Ken Jennings and Brad Rutter in 2011. The Watson avatar sat between the

two other contestants, as a human competitor would, while its considerable bulk sat on a different floor of the building. Like the other contestants, Watson had no Internet access.

In the practice round, Watson demonstrated a human-like ability for complex wordplay, correctly responding, for example, to “Classic candy bar that’s a female Supreme Court justice” with “What is Baby Ruth Ginsburg?” Rutter noted that although the retrieval of information is “trivial” for Watson and difficult for a human, the human is still better at the complex task of comprehension. Nevertheless, machine learning allows Watson to examine its mistakes against the correct answers to see where it erred and so inform future responses.

In an interview during the Jeopardy! practice round, an IBM representative evaded the question of whether Watson might be made broadly available through a Web interface. The representative said that the company was currently more interested in vertical applications such as healthcare and decision support.

However thinking different, Cognitive Computing systems will bring date-led change. Today's programming models are things of the past, says consultant Judith Hurwitz. Cognitive computing systems use data to devise applications that reflect the world as we see it, in particular, the way business applications are built today is flawed.

Judith Hurwitz, a longtime consultant and author of numerous books on IT, made the declaration at the recent Cloud Expo in New York. The building process is flawed, she said, because it relies on business logic - the programming that lays out what operations will be set in motion and what tasks will be done.

“We cannot continue to write programs the old-fashioned way -- with logic, with a beginning, a middle and an end -- and then feed in the data and everything will be OK,” Hurwitz said. “That's why we have so many problems as we do today with our systems. They were built on how we thought about a problem 20, 50, 5,000 years ago.”

Over the years, as business models have changed and the number of data sources multiplied, organizations tweaked their old programs to fit, patching the logic and creating what Hurwitz called “monster systems.”

Cognitive computing systems, which use hardware or software to approximate human cognitive functions, will change all that, Hurwitz said. The business applications of the future will be based on fast-moving, ever-changing data from an ever-growing number of sources. Gone will be step-by-step instructions based on the past. Cognitive computing learns from patterns and anomalies, makes guesses about what could happen - and it doesn't assume there is one correct answer. As more data is ingested and analyzed, the system changes, too.

Though cognitive computing is designed to operate in much the same way as a human brain does - absorbing a large amount of information, learning from it and making hypotheses based on it - science has yet to replicate the brain. So, cognitive computing systems - “unless we fast-forward 100 years when we can simulate all the synapses in the brain” - won't work on their own; they'll rely on collaboration between humans and machines, Hurwitz said. Humans bring with them the knowledge they already have, and machines bring the capacity to process and store much, much more of it.

5.14.1. Examples of Artificial Intelligence Technology

Few examples of Artificial Intelligence technology can be listed here that are ongoing process in present market and they listed here:

- Automation is the process of making a system or process function automatically. Robotic process automation, for example, can be programmed to perform high-volume, repeatable tasks normally performed by humans. Replication Protein A (RPA) is different from Information Technology (IT) automation in that it can adapt to changing circumstances.

What is Replication Protein A (RPA)

Replication protein A (RPA) is the major protein that binds to single-stranded DNA (ssDNA) in eukaryotic cells [1, 2]. In vitro, RPA shows a much higher affinity for ssDNA than RNA or double-stranded DNA.

During DNA replication, RPA prevents single-stranded DNA (ssDNA) from winding back on itself or from forming secondary structures. This keeps DNA unwound for the polymerase to replicate it. RPA also binds to ssDNA during the initial phase of homologous recombination, an important process in DNA repair and prophase I of meiosis.

Hypersensitivity to DNA damaging agents can be caused by mutations in the RPA gene. Like its role in DNA replication, this keeps ssDNA from binding to itself (self-complementising) so that the resulting nucleoprotein filament can then be bound by Rad51 and its cofactors.

Figure 5-31. An Image of Human Replication Protein A (RPA).

RPA also binds to DNA during the nucleotide excision repair process. This binding stabilizes the repair complex during the repair process. A bacterial homolog is called Single-Strand Binding protein (SSB).

- Machine learning is the science of getting a computer to act without programming. Deep learning is a subset of machine learning that, in very simple terms, can be, thought of as the automation of predictive analytics. There are three types of machine learning algorithms: supervised learning in which, data sets are labeled so that patterns can be detected and used to label new data sets. Unsupervised learning, in which data sets are not labeled and are sorted according to similarities or differences; and reinforcement learning, in which data sets aren't labeled but, after performing an action or several actions, the AI system is given feedback.

- Machine vision is the sciences of making computers see. Machine vision captures and analyzes visual information using a camera, analog-to-digital conversion and digital signal processing. It is often compared to human eyesight, but machine vision isn't bound by biology and can be programmed to see through walls, for example. It is used in a range of applications from signature identification to medical image analysis. Computer vision, which is focused on machine-based image processing, is often conflated with machine vision.
- Natural Language Processing (NLP) is the processing of human -- and not computer -- language by a computer program. One of the older and best known examples of NLP is spam detection, which looks at the subject line and the text of an email and decides if it's junk. Current approaches to NLP are based on machine learning. NLP tasks include text translation, sentiment analysis and speech recognition.
- Pattern recognition is a branch of machine learning that focuses on identifying patterns in data. The term, today, is dated.
- Robotics is a field of engineering focused on the design and manufacturing of robots. Robots are often used to perform tasks that are difficult for humans to perform or perform consistently. They are used in assembly lines for car production or by NASA to move large objects in space. More recently, researchers are using machine learning to build robots that can interact in social settings.

5.14.2. What Is a Chatbot?

Chatbot is a computer program designed to simulate conversation with human users, especially over the Internet. The term "Chatter Bot" was originally coined by Michael Mauldin (creator of the first Verbot, Julia) in 1994 to describe these conversational programs. A *chat bot* (also known as a *talk bot*, *chatter bot*, Bot, chatterbox, Artificial Conversational Entity) is a computer program which conducts a conversation via auditory or textual methods. Such programs are often designed to convincingly simulate how a human would behave as a conversational partner, thereby passing the Turing test. Chatter bots are typically used in dialog systems for various practical purposes including customer service or information acquisition. Some chatter bots use sophisticated natural language processing systems, but many simpler systems scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a database.

In 1950, Alan Turing's famous article "Computing Machinery and Intelligence" was published [2], which proposed what is now called the Turing test as a criterion of intelligence. This criterion depends on the ability of a computer program to impersonate a human in a real-time written conversation with a human judge, sufficiently well that the judge is unable to distinguish reliably—on the basis of the conversational content alone—between the program and a real human. The notoriety of Turing's proposed test stimulated great interest in Joseph Weizenbaum's program ELIZA, published in 1966, which seemed to be able to fool users into believing that they were conversing with a real human. However Weizenbaum himself did not claim that ELIZA was genuinely intelligent, and the Introduction to his paper presented it more as a debunking exercise:

There are two main types of chat bots, one functions based on a set of rules, and the other more advanced version uses artificial intelligence. The chat bots based on rules, tend to be limited in functionality, and are as smart as they are programmed to be. On the other end, chat bots that use artificial intelligence, understands language, not just commands, and continuously gets smarter as it learns from conversations it has with people.

In artificial intelligence: machines are made to behave in wondrous ways, often sufficient to dazzle even the most experienced observer. But once a particular program is unmasked, once its inner workings are explained its magic crumbles away; it stands revealed as a mere collection of procedures ... The observer says to himself "I could have written that." With that thought he moves the program in question from the shelf marked "intelligent," to that reserved for curios.

Now the question is why chatbots are such a big opportunity?. The answer is, because for the first time ever people are using messenger apps more than they are using social networks. The following Figure 5-31 shows the messaging apps have surpassed social networks

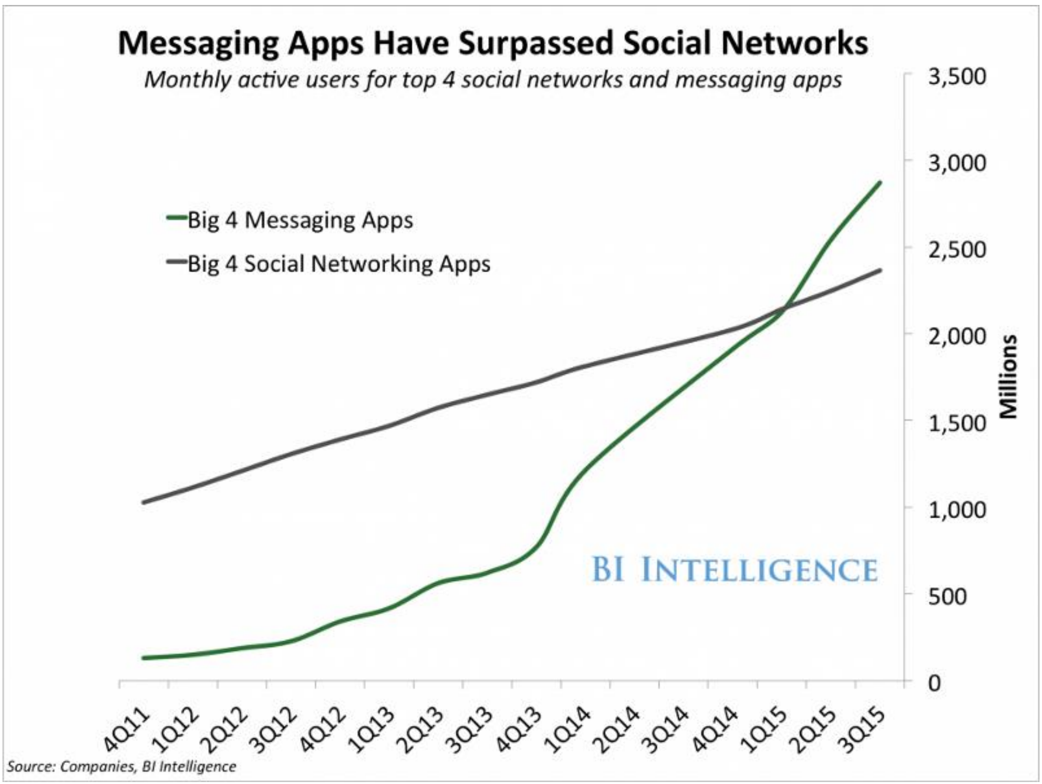


Figure 5-32. Messaging Apps Have Surpassed Social Networks.

People are using messenger apps more than they are using social networks.

> "People are now spending more time in messaging apps than in social media and that is a huge turning point. Messaging apps are the platforms of the future and bots will be how their users access all sorts of services."—Peter Rojas, Entrepreneur in Residence at Betaworks

So, logically, if you want to build a business online, you want to build where the people are. That place is now inside messenger apps.

> "Major shifts on large platforms should be seen as an opportunities for distribution. That said, we need to be careful not to judge the very early prototypes too harshly as the platforms are far from complete. I believe Facebook's recent launch is the beginning of a new application platform for micro application experiences. The fundamental idea is that customers will interact with just enough UI, whether conversational and/or widgets, to be delighted by a service/brand with immediate access to a rich profile and without the complexities of installing a native app, all fueled by mature advertising products. It's potentially a massive opportunity."—Aaron Batalion, Partner at Lightspeed Venture Partners

This is why chatbots are such a big deal. **It is potentially a huge business opportunity for anyone willing to jump headfirst and build something people want**.

But, how do these bots work? How do they know how to talk to people and answer questions? Isn't that artificial intelligence and isn't that insanely hard to do?

5.14.3. How Chatbots Work

There are two types of chatbots, one functions based on a set of rules, and the other more advanced version uses machine learning.

What does this mean?

Chatbot that functions based on rules:

- This bot is very limited. It can only respond to very specific commands. If you say the wrong thing, it doesn't know what you mean.
- This bot is only as smart as it is programmed to be.

Chatbot that functions using machine learning:

- This bot has an artificial brain AKA artificial intelligence. You don't have to be ridiculously specific when you are talking to it. It understands language, not just commands.
- This bot continuously gets smarter as it learns from conversations it has with people.

Bots are created with a purpose. A store will likely want to create a bot that helps you purchase something, where someone like Comcast might create a bot that can answer customer support questions.

5.14.4. Artificial Intelligence Applications

The following is wide range of artificial intelligence application in today's information technology cross enterprises and organizations.

- AI in healthcare. The biggest bets are on improving patient outcomes and reducing costs. Companies are applying machine learning to make better and faster diagnoses than humans. One of the best known healthcare technologies is IBM Watson. It understands natural language and is capable of responding to questions asked of it. The system mines patient data and other available data sources to form a hypothesis, which it then presents with a confidence scoring schema. Other AI applications include chatbots, a computer program used online to answer questions and assist customers, to help schedule follow-up appointments or aiding patients through the billing process, and virtual health assistants that provide basic medical feedback.
- AI in business. Robotic process automation is being applied to highly repetitive tasks normally performed by humans. Machine learning algorithms are being integrated into analytics and CRM platforms to uncover information on how to better serve customers. Chatbots have been incorporated into websites to provide immediate service to customers. Automation of job positions has also become a talking point among academics and IT consultancies such as Gartner and Forrester.
- AI in education. AI can automate grading, giving educators more time. AI can assess students and adapt to their needs, helping them work at their own pace. AI tutors can provide additional support to students, ensuring they stay on track. AI could change where and how students learn, perhaps even replacing some teachers.
- AI in finance. AI applied to personal finance applications, such as Mint or Turbo Tax, is upending financial institutions. Applications such as these could collect personal data and provide financial advice. Other programs, IBM Watson being one, have been applied to the process of buying a home. Today, software performs much of the trading on Wall Street.
- AI in law. The discovery process, sifting through of documents, in law is often overwhelming for humans. Automating this process is a better use of time and a more efficient process. Startups are also building question-and-answer computer assistants that can sift programmed-to-answer questions by examining the taxonomy and ontology associated with a database.
- AI in manufacturing. This is an area that has been at the forefront of incorporating robots into the workflow. Industrial robots used to perform single tasks and were separated from human workers, but as the technology advanced that changed.

5.15. The Robot Apocalypse

We did find an article on Internet that talks about the subject of “The Robots Apocalypse: Will Chatbots Make Call Center Agents Extinct,” We could not find who is the author of this article to give the proper credit to him or her, but we quote him or her here by reprinting the article to some degree. However it seems the www.247-inc.com is responsible for the article.

Stephen Hawking, Britain's pre-eminent theoretical physicist told the BBC in 2014, "The development of full artificial intelligence could spell the end of the human race. It would take off on its own, and re-design itself at an ever-increasing rate. Humans, who are limited by slow biological evolution, could not compete, and would be superseded" [25].

But before you start building a bunker and preparing yourself for the imminent robot apocalypse, you should know that today's artificial intelligence (AI) technology is a long way from the ominous prophecy Hawking warned us about. Current AI technology depends heavily on human assistance to be successful, and serves largely to make humans more effective and productive [26].

In the customer experience world, this human AI collaboration has taken the form of chatbots.

What exactly is a chatbot? In the simplest terms, a chatbot is a computer program designed to simulate conversation with human users, especially over the internet. Today's chatbots incorporate a wide range of services, from basic Q & A, to the more sophisticated self-service experience of virtual assistants [27].

And while today's chatbots are capable of a truly impressive range of interaction and decision making, you shouldn't fear—or plan—that chatbots will replace all your customer service agents. Like so much of AI technology, it's the combination of human interaction with chatbot technology that will maximize customer service effectiveness.

Although the chatbot buzzword seems to be on everyone's lips these days, a common fear among customer service leaders is that chatbots may create more problems than they solve by giving customers incorrect or irrelevant information. This is easily prevented. The key to successful implementation of AI lies in knowing how to strike the right balance between live, empathetic humans, and fast, low-cost, automated robots [28].

And that is where this report comes in. Think of it as your primer on the topic of AI and chatbot. Your shortcut for getting up to speed on how AI works, how chatbots use AI to emulate humans, and how AI and agents should be combined to deliver maximum benefit to the organization and the customer.

REFERENCES

[1] David Leech Anderson, "Humans using machines, humans as machines: Implications for teaching and learning," Humanities and Technology Review Fall 2008, Volume 27. Pages 1-23 ISSN 1076-7908.

[2] Davis Leech Anderson, http://www.mind.ilstu.edu/curriculum/modOverview.php?modGUI=239.

[3] Bahman Zohuri, Directed Energy Weapons: Physics of High Energy Lasers (HEL) 1st ed. 2016 Edition, Springer Publishing Company.

[4] David L. Anderson, http://www.mind.ilstu.edu/curriculum/functionalism_intro/functionalism_intro.php?modGUI=44&compGUI=1945&itemGUI=3403.

[5] http://www.mind.ilstu.edu/curriculum/nature_of_computers/computer-types.php?modGUI=196&compGUI=1747&itemGUI=3016.

[6] Gears of war: When mechanical analog computers ruled the waves Ars Technica.

[7] E. Beggs & J. Tucker (2006). 'Embedding infinitely parallel computation in Newtonian kinematics.' *Applied mathematics and computation* **178** (1):25–43.

[8] E. Beggs & J. Tucker (2007). 'Can Newtonian systems bounded in space, time, mass and energy compute all functions?'. *Theoretical Computer Science* **371** (1):4–19.

[9] O. Bournez & M. Cosnard (1995). 'On the computational power and super-Turing capabilities of dynamical systems'. Tech. Rep. 95-30, Ecole Normal Superior de Lyons.

[10] Bahman Zohuri and Masoud Moghaddam, Business Resilience System (BRS): Driven Through Boolean, Fuzzy Logics and Cloud Computation: Real and Near Real Time Analysis and Decision Making System 1st ed. 2017 Edition.

[11] Mohamad H. Hassoun, Fundamentals of Artificial Neural Networks, MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, First Edition.

[12] Andrew Friedman, "The Fundamental Distinction between Brains and Turning Machines," published paper with neuroscience special report, pp 28-33, 2002.

[13] Fei Su of Intel Corporation and Krishnendu Chakrabarty of Duke University, "High-Level Synthesis of Digital Microfluidic Biochips." Duke University.

[14] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," Science 270, pp. 467–470, 1995.

[15] G. MacBeath, A. N. Koehler, and S. L. Schreiber, "Printing small molecules as microarrays and detecting protein-ligand interactions en masse," J. Am. Chem. Soc. 121, pp. 7967–7968, 1999.

[16] Jitesh Dundas and David Chik, " Implementing Human-like Intuition Mechanism in Artificial Intelligence" Edencore Technologies Ltd. Row House 6, Opp Ambo Vihar, Tirupati Nagar-II, Virar (w), Thane-401303, India.

[17] Kahneman D. (2003) A Perspective on Judgment and Choice. American Psychologist, 58(9), 697-720.

[18] John McCarthy. Mathematical Logic in Artificial Intelligence. *Daedalus*. Vol. 117, No. 1. Artificial Intelligence (Winter, 1988), pp. 297-311.MIT Press on behalf of American Academy of Arts & Sciences. Online: - http://www.jstor.org/stable/20025149.

[19] Aaron Sloman. INTERACTIONS BETWEEN PHILOSOPHY AND ARTIFICIAL INTELLIGENCE: The Role of Intuition and Non-Logical Reasoning in Intelligence. Artificial Intelligence 2 (1971), 209-225.

[20] Jitesh Dundas and David Chik. IBSEAD: - A Self-Evolving Self-Obsessed Learning Algorithm for Machine Learning. IJCSET (URL: - http://ijcset.excelingtech.co.uk/). Volume 1. Issue 4. No 48. December, 2010.

[21] Jitesh Dundas. Law of Connectivity in Machine Learning. International Journal of Simulation- Systems, Science and Technology - IJSSST (URL: - http://www.ijssst.info/). Vol. 11, No. 5. Dec 2010. (ISSN: 1473-804 x Online) and (ISSN: 1473-8031 Print). UK.

[22] Christopher M. Bishop, "Patten Recognition and Machine Learning," Published by Springer Publishing Company, 2006.

[23] http://www.bbc.com/news/technology-30290540.

[24] http://searchcio.techtarget.com/news/4500260142/Despite-progress-the-future-of-AI-will-require-human-assistance.

[25] http://www.bbc.com/news/technology-30290540.

[26] http://searchcio.techtarget.com/news/4500260142/Despite-progress-the-future-of-AI-will-require-human-assistance.
[27] http://www.247-inc.com/company/blog/strike-balance-chatbots-vs-humans-telecom-landscape.
[28] http://venturebeat.com/2016/07/23/chatbots-will-take-over-customer-service-not-so-fast/.

Chapter 6

COMPUTATIONAL NEUROSCIENCE

The last several years have seen a dramatic increase in the number of neurobiologists building or using computer-based models as a regular part of their efforts to understand how different neural systems function (Eeckman and Bower 1993) [1], (Bower 1992) [2]. As experimental data continue to be amassed, it is increasingly clear that detailed physiological and anatomical data alone are not enough to infer how neural circuits work. Experimentalists appear to be recognizing the need for the quantitative approach to exploring the functional consequences of particular neuronal features that is provided by modeling. This combination of modeling and experimental work has led to the creation of the new discipline of computational neuroscience (Eeckman and Bower 1993) [1]. In addition, we have obtained permission from Professor David Beeman of University of Colorado to publish most of his lectures on the subject of Computational Neuroscience, here in this chapter word-by-word and take all his advice and recommendation to tailor them for this purpose.

6.1. INTRODUCTION

Neurocomputing is a subject that has captivated the interest of thousands of scientist, technologist, artificial design engineers, and mathematicians. The idea of training a system to carry out an information processing functions instead of programming it has intrinsic appeal. Perhaps, this is because of our own personal interest and familiarity with training as an easy and natural way to acquire new information processing capabilities. Neurocomputing or Computational folks, fall in love with systems that are often, endowed with a "look and feel" vaguely.

We can describe computational neuroscience or neurocomputing as a technological discipline concerned with information processing systems such as *neural networks* that autonomously develop operational capabilities in adaptive response to an information environment and this is capability that a Business Resilience System (BRS) requires to have [3].

Neurocomputing is a fundamentally new and different approach to information processing. It is the first alternative to *programmed computing*, which has dominated information processing for the last 50 years. Solving a problem using programmed computing involves devising an algorithm and/or Boolean type rule or logic or a set of rules for solving

the problem and then correctly coding these in software and making necessary revisions and improvements at the end.

Clearly, programmed computing can be used in only those cases where the processing to be accomplished can be described in terms of a known procedure or a known set of rules. If the required algorithmic procedure and/or set of rules are not known, then they must be developed -- an understanding that, in general has been found to be costly and time consuming. As matter of fact, if the algorithm required is not simple, which is frequently is the case for most desired capabilities; the development process may have to wait a flash of insight or several flashes.

Knowing what we know about computer programming in today's computation, such an innovation process cannot be accurately planned or controlled. Even when the required algorithm or rule set can be devised, the problem of software development still must be faced.

The Computer Aided Software Engineering (CASE) tools often used with neurocomputing systems can frequently be, utilized to build these routine software modules in a few short time periods. However, *neurocomputing* is the technological discipline concerned with parallel, distributed, adaptive information processing systems that develop information processing capabilities in response to exposure to an information environment [4].

The primary and essential information processing structures of interest in neurocomputing are *neural networks*, although other classes of adaptive information processing structures are sometime need to be, considered as well. This includes machine learning such as automation, genetic learning systems, data-adaptive content addressable memories, simulated annealing systems, associative memories, and fuzzy learning systems, based on fuzzy logics both first and second types.

Hecht-Nielsen has an excellent description for definition of neural network, which we have quoted here directly [4].

DEFINTION: A Neural Network is a parallel, distributed information processing structure consisting of processing elements which can possess a local memory and can carry out localized information processing operations interconnected via unidirectional signal channels called connections. Each processing element has a single output that branches ("fans out") into as many collateral connections as desired; each carries the same signal -- the processing element output signal. The processing element output signal can be of any mathematical type desired. The information processing that goes on within each processing element can be defined arbitrary with the restriction that it must be completely local; that is, it must depend only on the current values of the input signals arriving at the processing element via impinging connections and on values stored in the processing element's local memory.

Later on, in this chapter we put the above definition by Hecht-Nielsen in more perspective that is practical rather than being as abstract as it is describe by him.

6.2. OVERVIEW OF NEUROCOMPUTING

To illustrate the nature of neural networks we shall describe briefly a classical neural network architecture known as the *perceptron.* Because it has been largely superseded by more powerful neural networks, for example, the perceptron is primarily of historical interest,

although it is occasionally used. The perceptron is a neural network that consists of one or more of the processing elements shown in Figure 6-1, which are themselves also referred to individually as perceptrons.

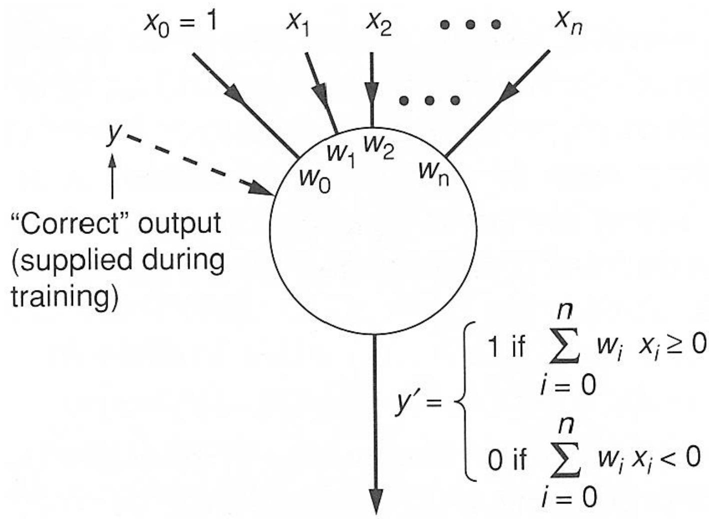


Figure 6-1. The Single Perceptron Processing.

For simplicity, we shall concentrate on the operation of a single perceptron processing element. The perceptron has an input consisting of an $(n+1)$ dimensional vector $\vec{\boldsymbol{x}} = (x_0, x_1, x_2, \cdots, x_n)$, where x_0 is permanently set to 1, which is called a *bias input.* The output of the perceptron is 1 if the weighted input sum $x_0 w_0 + x_1 w_1 + x_2 w_2 + \cdots + x_n w_n$ is greater than or equal to zero; the output is 0 if this weighted input sum is less than zero. However, the goal of the perceptron is illustrated in Figure 6.2.

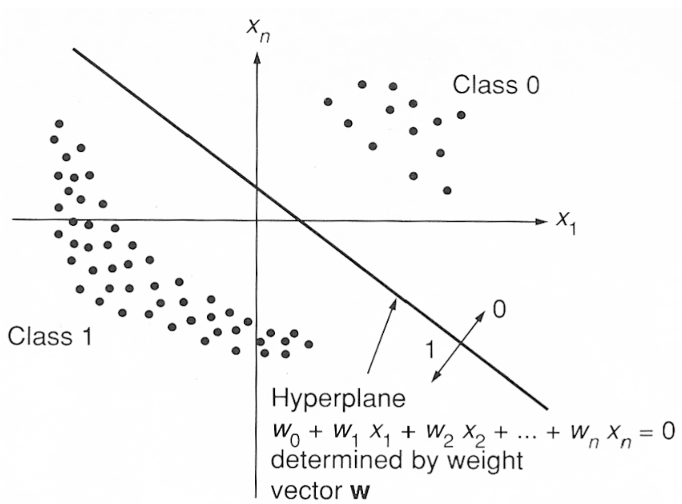


Figure 6-2. A Patten Classification Problem in n-Dimensional Space.

Here we see two classes of *patterns*, which are class 0 and class 1. A pattern is simply a point in n-dimensional space. The coordinates of the point represent *attributes* or *features* of the object to be classified, such as weight, height, density, or frequency. Basically, the goal is to find a set of weights for which the output of the perceptron will always match the class number of the point entered into the perceptron. The perceptron weight vector $\vec{w}$ determines a hyper-plane in n-dimensional space. If a point $(x_0, x_1, x_2, \cdots, x_n)$ lies on one side of hyper-plane marked with a "0", then the output of the perceptron is 0. If the point lies on the hyper-plane or on the "1" side of the hyper-plane, then the perceptron output is 1. In the situation illustrated in Figure 6-2, the perceptron will perform correctly because its hyper-plane has been oriented properly relative to the two linearly separable classes. Note that in 2-dimensional space, a hyper-plane is a line, in 3-dimensional space is an ordinary plane, and finally in n-dimensional space it is and $(n-1)$-dimensional flat surface. Classes that have this property are termed *linearly separable*. The objective is to find a set of *weight* or *adaptive coefficients* $w_0, w_1, \cdots, w_n$, which it turns out, determine a unique hyper-plane, such that the output of the perceptron is 1 if the input pattern vector $\vec{x} = (x_0, x_1, x_2, \cdots, x_n)$ belongs to class 1, and o if the pattern vector belongs to class 0.

Process-wise the weights are stored within the processing element and are automatically modified by processing element itself in accordance with the *perceptron learning law*. This learning law is then operates during process where the perceptron is shown a sequence of randomly selected $\vec{x}$ example is presented to the perceptron as part of a *training trial*, the system is also is instructed to which class 0 or 1 the example belongs. On each training trial, the learning law modifies the weight vector $\vec{w}$ in accordance with the following equation:

$$\vec{w}^{\text{new}} = \vec{w}^{\text{old}} + (y - y')\vec{x} \qquad \text{Eq. 6-1}$$

where y is the correct class number of the input pattern $\vec{x}$ which, is supplied, along with $\vec{x}$, on each training trial, and y' is the output of the perceptron. The idea of this learning law is that, if the perceptron makes an error $(y - y')$ in its output, however, this error is indicating that a need to reorient the weight vector $\vec{w}$ hyper-plane so that the perceptron will tend not to make an error on this particular vector $\vec{x}$ or any other vector near it again.

We should note that the output error $(y - y')$ will be 0 if the output of the perceptron is correct, however, in this situation the weight will not change. If the output is wrong, then $(y - y')$ will be either +1 or -1, and $\vec{w}$ will be modified appropriately so that the perceptron will do better in the future processing of the information.

Readers should look into history of perceptron that was originally invented in 1957 by Frank Rosenblatt [5], who also wrote about *"Principle of Neurodynamics,"* one of the two early books on neurocomputing [6]. Karl Steinbuch 7 writes the other book that you should know about it, with title of "*Automat und Mensch.*"

6.3. NEUROCOMPUTING AND NEUROSCIENCE

Neuroscience can be defined as the scientific discipline that as associated with understanding both the *Brain* and the *Mind*, which, are usually, presumed analogously to be, respectively, the "*hardware*" and "*software*" aspect of the same object, as we can see in a computer system. Clearly, we can consider the brain as wet computer and it is composed of network of neurons, however, these neurons are much more complicated than processing elements used in neurocomputing, and their functions, are not yet, understood to the perfection of their functionality and operation in our brain.

Progress in neuroscience like any other science has been, made by creating functional concepts and models, based upon experimental results and then refining or refuted, are shown to be excessively oversimplified. A good way to think of the brain is as an exceedingly complex object built using an alien technology so advanced that we are only now beginning to understand, after more than a century of concentrated study toward this object at the very simplest component level of it.

In neuroscience, the production of functional concepts and model, is continuous and prodigious, thus these will provide an excellent source of new concepts and principles as well as a foundation for use in neurocomputing and continue expanding going forward with time. Since these neuroscience concepts and models are *not* accurate representation as of yet for brain function, therefore, neurocomputing systems based upon these ideas cannot be, described as being "based upon operation of the human brain" or what we call, an ideal wet-computer. To be more precise, we can state, that neurocomputing systems based upon these ideas, probably have no close relationship to the operation of the human brain.

The stream of ideas from neuroscience to neurocomputing is only half of the fact, however there is also beginning to be a flow of ideas from neurocomputing to neuroscience as well. Neuroscientists and researchers belonging to neurocomputing community are constantly developing new neural network architectures in particular related to artificial aspect of the subject, or what we have started as foundation of writing this book. These architectural developments do include a new concepts and theories, which can explain the operation of these approaches.

Many of these developments can be used by neuroscientists as new paradigms for building function concepts and models of elements of brain and mind as part of new generation of Artificial Neural Network (ANN) and eventually the new generation of smarter robots, that can process the incoming information in real-time.

6.4. NEURAL NETWORKS CONCEPTS

Neural networks are fundamentally simple structures, however, as with almost any class of structured objects, such as automobile, aircraft, organization like infrastructure of governments, or for that matter any languages and their associates grammars, etc., it is inventible to have a detailed general of canonical model for the members of the class. Understanding of models detailed it will provide a path to prototype design and eventually to full production of desired objects [8].

Such model typically provides a set of descriptive terms such as dimensions and measurements that can be used to characterize the prototype and a particular member of the class. Further, down this chapter, we provide more details and useful facts and definitions concerning this subject.

To sum up this section, we need to point out again, neural networks are composed of processing elements and connections. This refined definition is then elaborates upon to form a general structural model of a neural network. The general model can be presented here, what is known as the *AXON* model. The AXON model not only provides a convenient way to specify the structure of neural network. It also provides us with a set of descriptive terms for neural networks that will be useful in the remaining of this chapter.

We continue this summary with Robert Hecht-Nielsen [4] definition of a neural network, as he describes it. A *direct graph* is a geometrical object consisting of a set of points that we call them *nodes* along with a set of direct line segments, which, is called *links* between them. A *neural network* is a *parallel* distributed information processing structure as we stated in previous chapters in the form of a directed graph, with the following sub-definitions and restrictions:

1) The nodes of the graph are called processing elements.
2) The links of the graph are called connections. Each connection functions as an instantaneous unidirectional signal-conduction path.
3) Each processing element can receive any number of incoming connections that is also called input connections.
4) Each processing element can have any number of outgoing connections, but the signals in all of these must be the same. In effect, each processing element has a single output connection that can branch or fan out into copies to form multiple output connections (sometimes called collaterals), each of which carries the same identical signal (the processing element's output signal).
5) Processing elements can have local memory.
6) Each processing element possesses a transfer function which can use (and alter) local memory, can use input signals, and which produces the processing element's output signal. In other words, the only inputs allowed to the transfer function are the values stored in the processing element's local memory and the current values of the input signals in the connections received by the processing element. The only outputs allowed from the transfer function are values to be stored in the processing element's local memory and the processing element's output signal. Transfer functions can operate continuously or episodically. If they operate episodically, there must be an input called “activate” that causes the processing element's transfer function to operate on the current input signals and local memory values and to produce an updated output signal (and possibly to modify local memory values). Continuous processing elements are always operating. The “activate” input arrives via a connection from a scheduling processing element that is part of the network.
7) Input signals to a neural network from outside the network arrive via connections that originate in the outside world. Outputs from the network to the outside world are connections that leave the network.

Figure 6-3 shows typical neural network architecture. In neurocomputing, the word architecture is reserved for the formal mathematical description of a neural network. Just as the definition of an algorithm in programmed computing has nothing to do with how that algorithm is run on a computer, so in neurocomputing the definition of neural network architecture has nothing to do with that architecture's implementation (meaning the manner in which a neural network is implemented in software, neuro-software, and/or hardware).

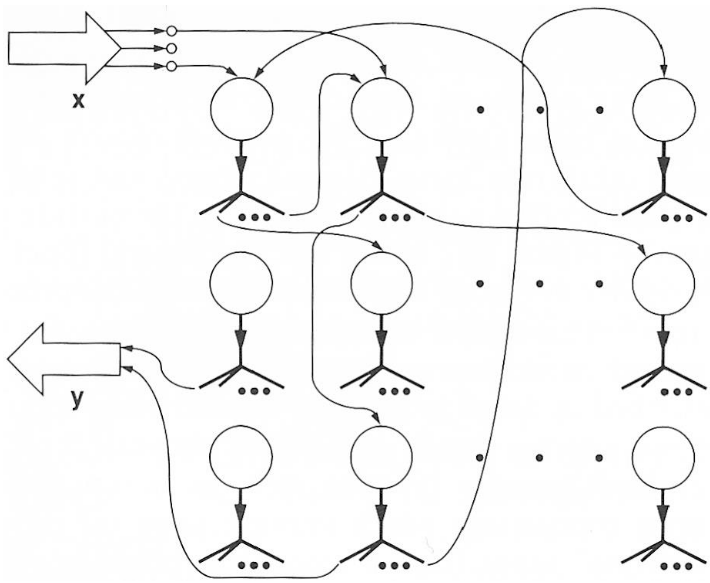


Figure 6-3. A Typical Neural Network Architecture.

Figure 6-3 illustrates some inputs, which, can be, thought of collectively as an input data array vector $\vec{\boldsymbol{x}}$ entering the network from the outside world and copies of selected processing element output signals, which, can be thought of collectively as an output data array vector $\boldsymbol{y}$ leaving the network and being supplied to the outside world [4].

In Figure 6-3, each processing element can have multiple input connections which, can originate from other processing elements or from outside the network, but only one output signal. This processing element is analogous to Markov Chain events, where a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.

The single output as shown in Figure 6-3, branches into copies, where multiple connections carrying the same signal which, are distributed to other processing elements, or which leave the network altogether. The input to the network can be reviewed in this way, the network can be thought of as a *function*, *subroutine,* or *procedure* $\boldsymbol{y}(\boldsymbol{x})$. This observation is the basis for the mechanism used to embed neural networks into programmed computing systems.

Processing elements within the network receive inputs from other processing elements and from these external inputs and send copies of their output signals to other processing elements and to the outside world [4].

In addition to the structure presented in above, all known neural networks have their processing elements divided into disjoint subsets, called *layers* or *slabs*, which mean exactly the same thing, in which all of the processing elements possess essentially the same transfer functions. In fact, this definition is universal for all neural networks because *any* neural network can be configured as a collection of layers; namely, by defining each layer to have a single processing element in it.

Figure 6-4 shows a neural network with six slabs. Each slab in the figure consists of a 2-dimensional array of processing elements. In addition, some useful facts and definitions concerning n-dimensional geometry can be illustrated as well [4].

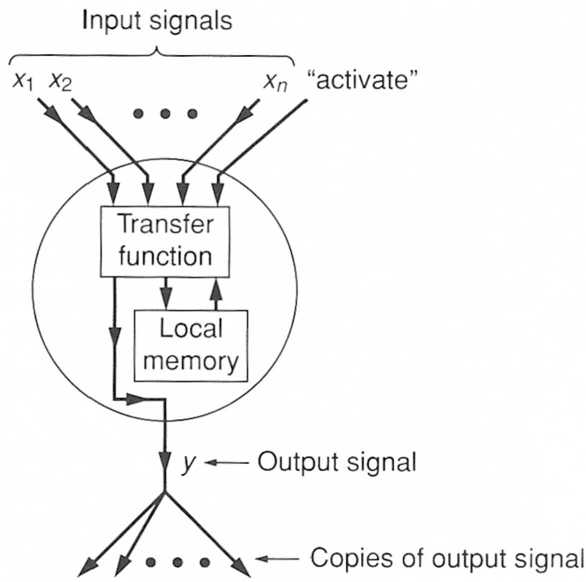


Figure 6-4. A Generic Processing Element.

Further analysis of illustration in Figure 6-4 revels that, the input signals $x_0, x_1, x_2, \cdots, x_n$ arriving at the processing element are supplied to the transfer function, as is the "activate" input. The "activate" input causes the transfer function to be activated. Continuous-time processing elements do not have an "activate" input -- their transfer functions are always active [4].

The transfer function of an episodically updated processing element, when activated, uses the current values of the input signals, as well as values in local memory, to produce the processing element's new output signal value y. The transfer function can also modify values stored in local memory to affect learning.

In case of Figure 6-4 illustration, slabs can have any geometrical form desired and processing elements can send connections to other processing elements on the same slab, as well as to processing elements on other slabs.

Many neural networks include a type of slab, called an *input slab*, in which each processing element receives exactly one input, which arrives from the outside world. Input slabs have their incoming signals defined by an array of data supplied by an external agent -- such as a computer program running on a host computer. The processing elements of the

input slab typically have no function other than to distribute the signals impinging upon them to other processing elements of the network.

As we have stated in above, such processing elements of network are, called *input units* or *fanout units*. They have no local memory and their transfer function is simply a latch, which releases the input signal to the outgoing collateral connections of the unit whenever the “activate” signal is received [4].

Processing element transfer functions usually have a sub-function, called a *learning law*, that is responsible for adapting the input-output behavior of the processing element transfer function (over a period of time) in response to the input signals that impinge on the processing element. This adaptation is usually, accomplished by modification of the values of variables stored in the processing element's local memory [4].

Not all neural network adaptation and learning take place via modification of values stored in local memory. Other possibilities exist. For example, connections between processing elements can be, created or destroyed. In other schemes, a new one might replace the transfer function of a processing element. The AXON neural network model presented here can typically accommodate these sorts of ideas. If the AXON model cannot accommodate an idea then it probably represents a process that is not compatible with neural networks. In this event, some other, more general, and information processing architectural context such as the class of general dataflow architectures or even the class of general MIMD architectures should be, considered.

In computing, MIMD (multiple instructions, multiple data) is a technique employed to achieve parallelism. Machines using MIMD have a number of processors that function asynchronously and independently. At any time, different processors may be executing different instructions on different pieces of data. See Figure 6-5.

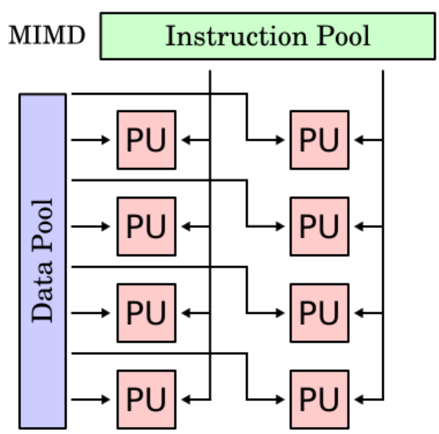


Figure 6-5. Illustration of a MIMD.

MIMD architectures may be, used in a number of application areas such as computer-aided design/computer-aided manufacturing, simulation, modeling, and as communication switches. MIMD machines can be of either shared memory or distributed memory categories.

These classifications are, based on how MIMD processors access memory. Shared memory machines may be of the bus-based, extended, or hierarchical type. Distributed memory machines may have hypercube or mesh interconnection schemes.

For further details, we encourage, the reads to refer to the book by Robert Hecht-Nielsen, "Neurocomputing" [4].

6.5. Fuzzy and Neural Networks

Artificial Neural Networks (ANN) and Fuzzy Logic (FL) work together, artificial neural networks classify and learn rules for fuzzy logic and fuzzy logic infers from unclear neural network parameters. The latter is a network with fast learning capabilities that produces intelligent, crisp output from fuzzy input and/or from fuzzy parameters and avoids time-consuming arithmetic manipulation.

Incorporating fuzzy principles in a neural network gives more user flexibility and a more robust system. Fuzziness in this case means more flexibility in the definition of the system; boundaries may be, described more generally, not crisply; inputs may be, described more vaguely, yet better control may be, obtained. The network itself may be fuzzy, not well defined, and able to reconfigure itself for best performance. The power of such machines may be, illustrated with the following "gedanken" examples.

Visualize a machine that has learned to analyze scenery, animals, other machines, and other items. A user describes a vague scene in terms of features such as "something as a tree, about here" and "something like an animal with four legs and a long tail and so tall, there," and so on. Then the machine draws a three-dimensional landscape with a tree and a dog nearby (and perhaps a mountain in the background, with a lake). Then the user may instruct the machine to make corrections to this scene, again in vague language, and the machine immediately projects a three-dimensional scene, very similar to the one the user had in mind. As all that is, done, a train with a whistling sound may be crossing the scene (if the parameters are set right) and nearby a frightened bird flies away.

Imagine a machine that is instructed to design a new three-dimensional' machine, based on some approximate specifications. Our gedanken machine designs a model from the vague specifications, simulates the created machine, makes corrections on the model, and, if the corrected one performs as expected, manufactures the first prototype-all in just a few minutes! [9].

6.6. Where are Fuzzy Neural Networks Heading?

Fuzzy logic follows the same path as Boolean and multiple value logic. Initially, binary logic started as a linguistic set of statements, such as if **A** = **B**, and if **B** = **C**, then is **A** = **C**? Then mathematical notation translated the linguistic statements into equations and theories were developed that are taught today. These theories have been, applied successfully in the development of many logical applications.

Fuzzy logic also started as a linguistic set of statements. For example, if A is taller than B, B is shorter than **C**, what is **A** with respect to **C**?. **A** number of mathematical theories can

be found in the literature. Thus, we may make a reasonable extrapolation and deduce that fuzzy logic will prove itself as binary logic did. The fusion of fuzzy logic and neural networks combines the best of each. Fuzzy concepts fused with “thinking” promise superior technology. These claims are, validated by various integrated circuits, fuzzy controllers for general applicability, and applications for automobile engine control, robot control, cameras (film and video), appliances, and the military. In addition to hardware solutions, numerous “fuzzy algorithmic solutions” have been applied in communications, signal processing (speech, image), and other areas. The number of companies relying on fuzzy logic is growing rapidly. Many significant American, European, and Asian-Pacific companies have announced products or are exploring and advancing fuzzy logic for potential applicability in their own products. In the near future, we will see applications that encompass algorithmic fuzzy logic, fuzzy neural networks, and combination of fuzzy and/or neural networks with high-performance microprocessors.

6.7. An Approach to Computational Neuroscience; a Reverse Engineering the Brain

There are various approaches to computational neuroscience, we might take, for example, the high-level systems approach - we can treat it as a black box and identify its input/output relationships. This corresponds to the psychological approach: the study of “mind.” Nevertheless, sooner, or later, we have to understand the hardware. We can try some circuit tracing and try to construct a wiring diagram. Unless it is a primitive single layer circuit board, this is hard - we may get incomplete knowledge. If we are lucky, we might be able to get some logic analyzer probes into critical pathways and try to figure out something from the relationships between the different pulse trains that we see.

Therefore, we will have to understand what the components do and how they work before we can understand their role in the circuit. As much as possible, we would like to understand the operation of simple sub-circuits, such as the clock generator before we understand the whole system. This reductionist approach is basically, the one we take in trying to understand the nervous system. We try to understand how the parts work and what they do, in order to understand the system as a whole.

In the case of the brain, we have several tools available, each with its limitations:

- Imaging techniques {Positron Emission Tomography (PET) scans, Magnetic Resonance Imaging (MRI), etc.) or Electroencephalography (EEG) - These have comparatively low resolution, but can help us create a “block diagram.”
- Neuron staining techniques - the technique goes back to Ramon y Cajal in the late 19th century - we can identify components and connections, to some extent, but we may have to make shrewd guesses and look for ways to test them.
- Confocal microscopy gives us a 3-D view of individual neurons
- Radioactive tracers can map axonal connections
- Intra and extracellular recording - simultaneous multicellular recording is analogous to using a logic analyzer
- Patch clamp techniques allow recording from individual ion channels

- Voltage sensitive dyes that change color according to the membrane potential, or dyes that are sensitive to the concentration of calcium ions.

The technique we would like to talk about is biologically realistic computer simulation. This approach falls under the heading of “Computational Neuroscience,” and it uses and complements information obtained from the techniques listed above.

However, computational neuroscientists often disagree about the amount of biological realism that is required. Why make detailed biologically realistic models, rather than simpler abstract models that try to get right to the important behavior of the system of interest? The brain has trillions of neurons, with complicated branching dendrites, and dozens of different types of ion-selective channels. It's a natural reaction to fear that if we get too bogged down in the details, we'll spend years trying to understand calcium diffusion in dendrites, or the behavior of some esoteric type of channel, and never get to the goal of “modeling the brain.”

It is tempting to make high-level abstract models without spiking neurons, hoping to discover some general principles of “how the brain computes.” For example, some people use mathematical models that treat the cortex as a collection of coupled oscillators, or use greatly simplified neurons that sum up their inputs and produce a binary output based on some simple criterion.

The problem with these sorts of models is that with enough ingenuity and adjustable parameters, you can usually construct a model with any given desired behavior. If you construct an artificial neural network that performs well at recognizing human faces, does this tell you anything about how your brain recognizes faces? If the model is detailed and biologically realistic, then experiments on the actual living system can greatly narrow down the choices that are, made in creating the model.

There is another trap that it is easy to fall into when deciding to use computer models. An obvious way to use modeling is to construct a model that incorporates some particular hypothesis and do “computer experiments” on the model in order to see if it behaves like experiments on the biological system. If it does, you might claim that this is evidence in favor of your hypothesis. The trouble with this is that it suffers from some of the same problems as abstract models: the simulation may be just giving the results that it was designed to give, and you don't know if a different model might have done just as well. With enough assumptions and tweaking of parameters, there are lots of models that could generate reasonable agreement with experiment.

A better approach is to try to ignore your preconceived ideas about the cause of a particular behavior and try to build the best model of the system you are studying that you can. This means incorporating the best physiological data that you can get to model the system in detail. This often means modeling neurons down to fine details of dendritic structure, and modeling each kind of ion channel that is, known to exist in the cell. This fosters a close relationship with experiment, because you will soon discover experiments that you need to do in order to get data to use to characterize some channel or membrane parameter.

Once you have done this, you often find that you have to fill in the gaps in your knowledge with some hypotheses. You might postulate some connections between neurons, which you feel might be necessary; or the existence of some interneuron in order to provide a needed inhibitory input. Or, you might assume the existence of some type of ionic channel which has been observed in other types of neurons and which seem necessary to explain the

behavior of the one you want to model. Then you use your simulation as a sort of breadboard to test out your ideas. If the new feature gives better agreement with experiment, then you are motivated to perform experiments to see if it really exists.

Of course, there will always be some parameters that, you have to estimate. Then you compare the behavior of the model with some “function-neutral” experiments like voltage or current clamp on a cell, or perhaps an electric shock to a nerve in a system. You can then “tune” the model by fitting any unknown parameters with data that is not directly related to the behavior of interest. If the model passes these tests, then you can have more confidence in it. At this point, you can start exploring the more interesting behavior of the model and understanding how the model produces this behavior. You can perform measurements and experiments on your model, which might be impossible in the real system. Everything is accessible in a simulation. You can try to simplify your model in order to find out what features are important for the interesting behavior of the system - and which features are merely “icing on the cake.”

You may discover that not only does the model behave like the biological system, but that the mechanisms that you postulated are causing that behavior. However, it is just as interesting to find that there is another cause, or even to see behavior in the model that you never thought to look for. This can be another useful way to guide the direction of experiments. This kind of realistic modeling opens up a new way to discover, through “computer experiments” the way that neurons process information.

6.7.1. Modeling Neurons

Now question is, how do we model a real neuron like this pyramidal cell in Figure 6-6, or a network composed of them in the hippocampus?

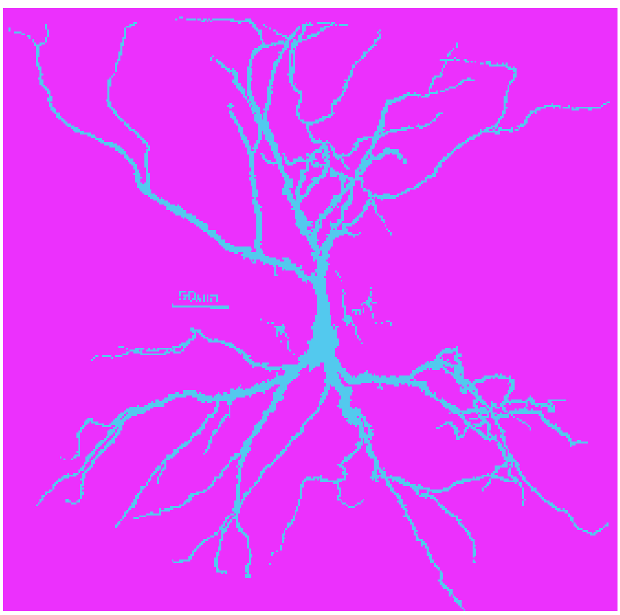


Figure 6-6. Neuron of Pyramidal Cell.

As you can see, it does not look much like the ones, you see in treatments of artificial neural networks, where you have a little circle with a summation sign inside. There are extensive trees of dendrites with the apical dendrites at the top, and basal dendrites at the bottom. Other neurons make synaptic connections at various points on this structure, and release neurotransmitters that open channels, allowing ions to flow in or out of the cell, resulting in the production of post-synaptic potentials. It turns out that these dendrites play an important role in the processing of information by the cell. The pyramid shaped cell body, or soma, contains voltage activated sodium and potassium channels somewhat similar to those studied by Hodgkin and Huxley in the giant axon of the squid. Post-synaptic potentials produced in the dendrites can propagate to the soma to trigger action potentials. Throughout the cell, we also have passive channels that remain partly open all the time, leading to a leakage resistance. The insulating cell membrane separates the conductive cytoplasm inside the cell from the salt-water environment outside, giving rise to a membrane capacitance. As the cytoplasm has some resistance, we also have an axial resistance along the dendrites and axon (not visible in this picture). Thus, a section of dendrite acts like a leaky cylindrical capacitor, coupled to neighboring sections with resistances.

In summary, we have a continuous distribution of resistance and capacitance across the membrane as well as an axial resistance parallel to the membrane. The various ion-selective channels across the membrane act like variable resistances.

The answer to our question is - we model it piece by piece. The usual approach is to model this with a lumped parameter model in which we divide the neuron into a finite number of compartments containing resistances, capacitances and batteries to represent ionic equilibrium potentials.

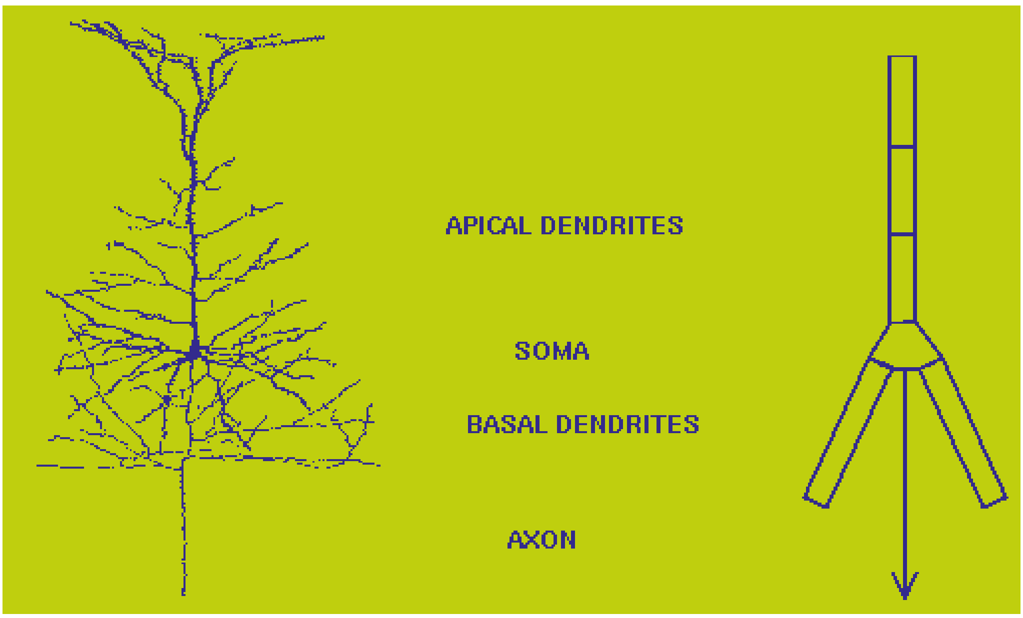
As you can see, it does not look much like those ones, you see in treatments of artificial neural networks, where you have a little circle with a summation sign inside. There are extensive trees of dendrites with the apical dendrites at the top, and basal dendrites at the bottom. Other neurons make synaptic connections at various points on this structure, and release neurotransmitters that open channels, allowing ions to flow in or out of the cell, resulting in the production of post-synaptic potentials. It turns out that these dendrites play an important role in the processing of information by the cell. The pyramid shaped cell body, or soma, contains voltage activated sodium and potassium channels somewhat similar to those studied by Hodgkin and Huxley in the giant axon of the squid. Post-synaptic potentials produced in the dendrites can propagate to the soma to trigger action potentials. Throughout the cell, we also have passive channels that remain partly open all the time, leading to a leakage resistance. The insulating cell membrane separates the conductive cytoplasm inside the cell from the salt-water environment outside, giving rise to a membrane capacitance. As the cytoplasm has some resistance, we also have an axial resistance along the dendrites and axon (not visible in this picture). Thus, a section of dendrite acts like a leaky cylindrical capacitor, coupled to neighboring sections with resistances.

In summary, we have a continuous distribution of resistance and capacitance across the membrane as well as an axial resistance parallel to the membrane. The various ion-selective channels across the membrane act like variable resistances.

The answer to our question is - we model it piece by piece. The usual approach is to model this with a lumped parameter model in which we divide the neuron into a finite number of compartments containing resistances, capacitances and batteries to represent ionic equilibrium potentials.

Figure 6-7(a) shows an example neuron based on a drawing of a pyramidal cell by Ramony Cajal that we would like to model, either as a single cell, or as a component in a network of interacting neurons. This figure shows the tree-like structure of the dendrites, which receive synaptic inputs from other neurons. Synaptically activated ion channels in the dendrites create postsynaptic potentials that, we assume here for simplicity, are passively, propagated to the pyramid-shaped cell body (soma) where voltage-activated ion channels may create action potentials. In most cells, these channels are concentrated near the base of the soma in the region called the axon hillock near the axon. The long axon at the bottom of the figure propagates action potentials to terminal branches that form synapses with other neurons.

In some cases, neurons may have voltage-activated channels in their dendrites. This not only complicates their electrical properties and thus their simulation, but also is responsible for the complex dynamics of these neurons.



(a) (b)

Figure 6-7. (a) A Pyramidal Cell with Dendrites, Soma, and Axon. (b) A Simplified Discrete Compartmental Model of the Same Neuron.

6.7.2. Detailed Compartmental Models

When constructing detailed neuronal models that explicitly consider all of the potential complexities of a cell, the increasingly standard approach is to divide the neuron into a finite number of interconnected anatomical compartments. Figure 6-7(b) shows a simplified model in which the neuron is, divided into several dendrite compartments, a soma, and an axon. Each compartment is then, modeled with equations describing an equivalent electrical circuit (Rall 1959) [10]. With the appropriate differential equations for each compartment, we can

model the behavior of each compartment as well as its interactions with neighboring compartments.

In this type of detailed compartmental model, each compartment must be, made small enough to be at approximately the same electrical potential. Often this means constructing simulations out of very large numbers of compartments. As a contrast to this simple model with just a few compartments, here is a model of a Purkinje cell from the cerebellum, constructed with the GEneral NEural SImulation System (GENESIS) simulator software by De Schutter and Bower (1994a) [11], (1994b) [12]. It has 4550 compartments and 8021 active conductances ("channels"). If you look closely, you can see that it is actually composed of many cylinders. The representation of this model is, shown in Figure 6-8.

Why would one need to construct such a detailed model? One reason is that, if we ultimately want a simple model, it is better to throw out details after proving that they are not significant, rather than just hoping that something important was not, omitted at the outset.

Sometimes this will lead to results that you would not have expected. Here is an example from a computer simulation of the Purkinje cell, where false color is being, used to represent the membrane potential throughout the cell. We can use this to find the effect of applying synaptic input at various places on the cell.

It has been, assumed for many years that dendrites process information "locally," and that very-local spatial, and temporal patterns of activity sum, in some complicated way, to produce cell output.

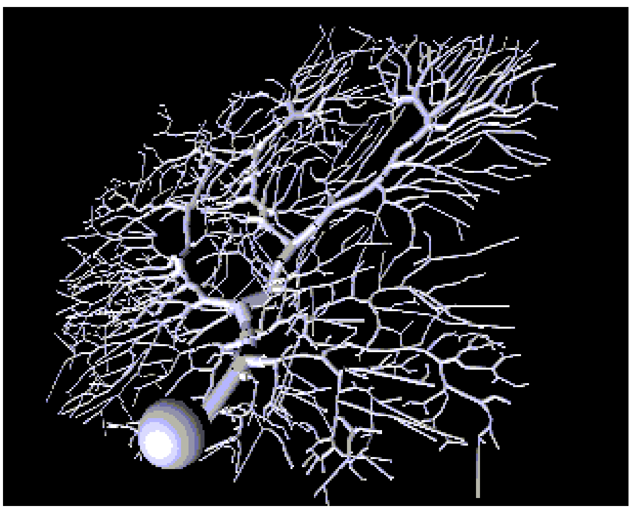


Figure 6-8. A detailed multi-compartmental model of a cerebellar Purkinje cell, created with GENESIS by De Schutter and Bower (1994a,b) [11-12]. Visualization by Jason Leigh using the GENESIS Visualizer program. The experimental data describing the cell morphology was provided by M. Rapp, I. Segev and Y. Yarom [13].

It was, believed that inputs far from the soma would have comparatively little effect, because post-synaptic potentials would be greatly attenuated as they pass through the axial resistance of the dendritic tree to the soma, as it is illustrated in Figure 6-9 in color. Modeling the cerebellar Purkinje cell suggests that the dendrite is actually operating globally - all regions of the dendrite have equal access to the soma, not as is usual, regions closer to the soma having a much larger influence. This comes about because a type of calcium channel

found in the dendrites (p-type), amplifies granule cell inputs more at the distal dendrites than at proximal dendrites. This process would be very hard to understand, or even to predict, without a good computer model. It is also, with the hindsight provided by the modeling results, what one would want to have in a Purkinje cell, which must receive and process as many as 100,000 synaptic inputs.

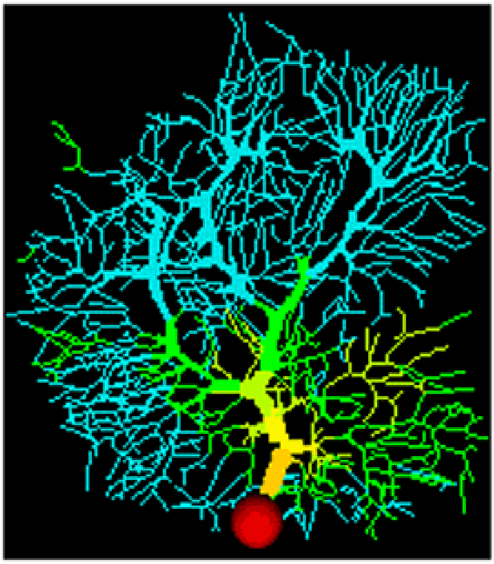


Figure 6-9. The Dendritic Tree Illustration.

6.7.3. Equivalent Cylinder Models

For some purposes, it may be adequate to model neurons with a smaller number of nonequipotential compartments. Models of this sort can be, used to model basic electrical properties of cells, or to construct small networks of neurons. Under these conditions, there are defined methods for constructing neuronal models dependent on the anatomy and physiology of the neuron in question. Many of these have been, pioneered by Wilfred Rall (cf., Segev, Rinzel and Shepherd 1995) [14]. For example, Rall has shown analytically that if dendritic trees approximately satisfy a “3/2 Power Law” and do not contain active conductances, they can safely be mapped into an equivalent linear structure. In this way, under defined conditions, a complicated branching structure can be approximated by a much simpler linear dendrite model, as was done in the model shown in Figure 7-6(b). However, in general, as the known complexity of the physiology or anatomy of the neuron increases, it is usually necessary to revert to full-blown compartmental models.

6.7.4. Single and Few Compartment Models

In cases where large numbers of neurons are being placed in network models, limited computer resources sometimes require that neurons be modeled with single compartments or

a very small number of compartments. For example, a large-scale GENESIS simulation of the olfactory cortex uses a network of 4500 neurons containing simple model pyramidal cells similar to the one shown in Figure 6-7(b) (Wilson and Bower 1989, 1992) [15]. As you may see for yourself, even such simplified neurons can sometimes capture experimentally observed behavior. We can see that only a single compartment is needed to model the behavior of some invertebrate “pacemaker” neurons. On the other hand, when building models of this type, one must always be aware that there are many local “computations” that occur in the extensive dendritic system of many neurons. Again, if these are of interest to the modeler, it is usually necessary to use hundreds or thousands of compartments.

6.7.5. Equivalent Circuit of a Single Compartment

Having described the general approaches to modeling single neurons, we now discuss in a bit more detail the basis for compartmental modeling. The reader should note that this section is intended as a very basic overview of neural modeling.

As we have described, the notion of an equivalent electrical circuit for a small piece of cellular membrane is the basis for all compartmental modeling. This arises from the fact that neuronal membranes have been, demonstrated to behave as simple electrical circuits with some capacitance, resistance, and voltage sources. These model parameters define the so-called *passive properties* that are responsible for the way that electrical impulses are, transmitted along the dendritic tree. It is generally necessary, as well as advisable, to begin all single cell modeling efforts with a consideration of passive cellular properties of the cell. These properties form the basis for the usually more interesting neuronal behavior that arises from the *active properties* provided by different voltage- or ligand-dependent conductances. If the passive properties are not modeled correctly, spurious results with active conductances are likely to be, obtained.

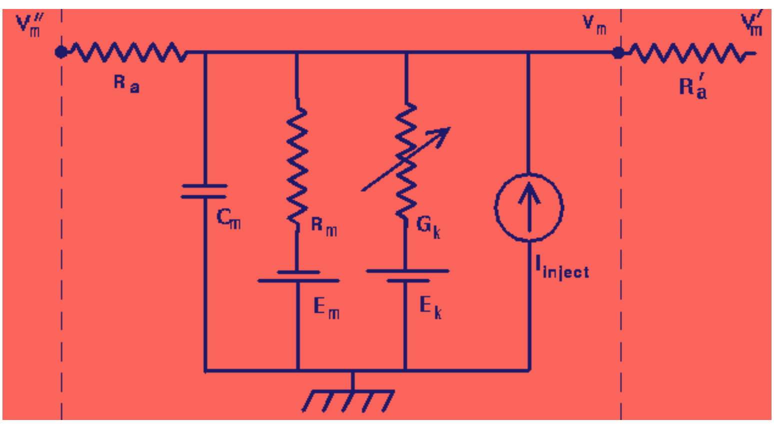


Figure 6-10. The Equivalent Circuit for a “Generic” Neural Compartment.

Figure 6-10 shows the equivalent electrical circuit of a basic neural compartment. Here, V_m represents the membrane potential, or the potential in the interior of a compartment relative to a point outside the cell. The “ground” symbol at the bottom of the figure represents

this external point, taken to be at zero potential. As the conducting ionic solutions inside and outside of the cell are, separated by the cell membrane, the compartment acts as a capacitor.

This is, charged or discharged by current flowing into or out of the compartment. This current flow may be from adjacent compartments, from the passage of ions through channels in the cell membrane, or from current injection from an electrode inserted into the cell.

The membrane potential appears across the membrane capacitance C_m, and can cause a current flow into or out of the compartment at the left through the axial resistance R_a when there is a difference in potential $V_m - V_m^{''}$ between the two compartments. Likewise, there may be a flow of current into or out of the primed compartment at the right through its axial resistance R_a.

The resistor with the arrow through it represents one of many possible variable channel conductances that are specific to a particular ion or combination of ions that give individual neurons and neuron types their unique computational properties. By convention, these are, described in terms of the conductance G_k rather than the resistance. As the conductance is the reciprocal of resistance, the units of G_k are in reciprocal ohms, or Siemens. Differences in the concentration of the ion between the inside and the outside of the cell result in an osmotic pressure, which tends to move ions along the concentration gradient. The resulting charge displacement creates a potential difference that opposes this flow. The membrane potential at which there is no net flux of the ion is the equilibrium potential (or *reversal potential*) E_k, represented by a battery in series with the conductance. In the absence of synaptic input, current injection, or spontaneous firing of action potentials, V_m will approach a steady state *rest potential* E_{rest}, typically in the range of -40 to -100 mV. This is, determined by the condition that there is no net current flow into the cell from the various types of ion channels.

The other resistor and battery link the exterior and the interior of the cell represent the combined effect of passive channels (mainly those for chloride ions) having a relatively fixed conductance. The resistance is usually, called the *membrane resistance* R_m, although it is sometimes referred to as a *leakage conductance* $G_{\mathrm{leak}} = 1/R_m$. The associated equilibrium potential E_m is typically close to the rest potential. In some cases, it is given a slightly different value, E_{leak}, in order to reduce the net channel current to zero when $V_m = E_{\mathrm{rest}}$. Finally, the current source I_{inject} represents an optional injection current, which could be, provided by an electrode inserted into the compartment.

One may then calculate V_m using a differential equation, which expresses the fact that the rate of change of the potential across C_m is proportional to the net current flowing into the

compartment to charge the capacitance. On the right-hand side of Equation 6-2, Ohm's law is, used to calculate the current due to each of the sources shown in Figure 6-10:

$$C_m \frac{dV_m}{dt} = \frac{(E_m - V_m)}{R_m} + \sum_k [(E_k - V_m) G_k] + \frac{(V_m^{'} - V_m)}{R_a} + \frac{(V_m^{''} - V_m)}{R_a} + I_{inject} \qquad \text{Eq. 6-2}$$

Here, the sum over *k* represents a sum over the different types of ion channels that are present in the compartment. The sign convention used in the GENESIS (The GENESIS book with collaboration of Professor David Beeman of University of Colorado, who helped us to write this chapter is under consideration and will be out soon) simulator defines a positive channel current to be one that causes a flow of positive charge *into* the compartment. The variable conductance of each channel type G_k, gives the net effect of many individual channels that open and close in a binary manner.

To model this on a computer, we need to numerically, solve Equation 6-2 for each compartment. Of course, the $V_m^{''}$ and $V_m^{'}$ in the adjacent compartments affect the currents flowing into or out of the compartments, so we are solving many coupled equations in parallel. In addition, we will need good models for the way that the conductances vary with voltage, time or synaptic input.

In addition, we will need good models for the way that the conductances vary with voltage, time or synaptic input. Usually, we can treat an axon as just a delay line for the propagation of action potentials, although it could also be, modeled as a series of compartments if we were interested in understanding the details of axonal propagation.

Hodgkin and Huxley used this approach to model a single compartment representing a short piece of squid giant axon. Since they did their work in the early '50s, they used hand crank mechanical calculators for the numerical integrations. We have a lot better tools now, and can solve harder problems, but the technique is essentially the same. If you have some programming experience, it would be fairly, simple for you to duplicate their model by writing a program in C, Java, or FORTRAN to solve the equations given in their papers. By using one of the freely available libraries for solving ordinary differential equations, you could even write your own neural simulator.

However, there are many advantages to using a general-purpose neural simulator and a high level simulation language, rather than writing your own simulation code in a computer programming language.

Built in tools - you do not have to write your own graphics display routines

- Designed to make it easy to add or subtract elements from your simulation - you do not have to go in and hack your code every time you make a change in your model.
- A good simulator will also have a library of simulation components like neural compartments, ion channels, synapses, and voltage-clamp circuit components as well as other objects, which you can use in your simulations.

A lot of early neural modeling was, done with SPICE - a general-purpose simulator for electronic circuits. Now there are simulators that have been, designed specifically for biologically realistic neural modeling.

These examples were created with GENESIS ((This book will be out soon), which was developed specifically for this type of modeling.

Hodgkin and Huxley's model (See next section) is at the basis of most single cell neuronal models. Most neurobiologists accept the importance of Hodgkin and Huxley's work and their development of the voltage clamp technique without recognizing how important the modeling was to the work. Essentially, the model was what made them throw out their old way of looking at the changes in the membrane and introduce the new one. It is important to remember that at the time of their experiments, the modern concept of ion-selective channels controlling the flow of current through the membrane was only one of several competing hypotheses. Their model ruled out these alternative ideas, and predicted the results of experiments as well, that were not used in formulating the model.

6.8. THE HODGKIN-HUXLEY MODEL

The diagram in Figure 6-11, illustrates the ionic basis of the neuron membrane potential.

Figure 6-11. Ionic Basis of the Neuron Membrane Potential.

It is, known that the concentration of sodium ions is greater outside the cell, so they have a tendency to enter. The concentration of potassium ions is greater on the inside, so they tend to leave. Metabolic processes within the cell called ionic pumps maintain this balance of concentrations. With this difference in ionic concentrations, the interior of the cell is, polarized to a membrane potential of about 70 mV negative to the outside, when the cell is at rest. Because of the higher exterior concentration of sodium, it has an equilibrium potential (reversal potential) of about 45 mV positive with respect to the cell, meaning that it will tend to enter the cell as long as the membrane potential is less than E_{Na}. Likewise, this competition between osmotic and electrostatic forces means that potassium will tend to leave unless the membrane potential falls below E_K.

Hodgkin and Huxley quantitatively explained the process by which depolarization of the cell (an increase in V_m) causes Na-selective channels to open, allowing Na ions to enter. This raises the potential, further increasing the conduction of the Na channels, and causes the

neuron to fire an action potential. Eventually, the Na channels inactivate, and K-selective channels begin to open, causing a flow of charge out of the cell, ending the action potential.

Now, let us look at the mathematical model that describes this behavior. As before, we start with a circuit diagram as illustrated, in Figure 6-12. In this case, it is for a piece of squid giant axon.

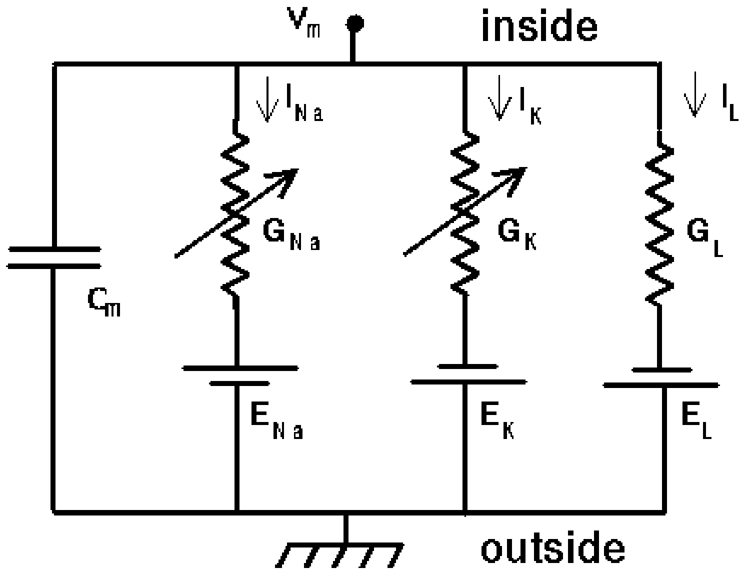


Figure 6-12. Illustration of Circuit Diagram.

Each of these two conductances represents the average effect of the binary gating of many ion channels.

The equation for I (in) - I (out) is similar to the one for the "generic compartment," but this is an isolated compartment, representing a piece of axon with the ends tied off in a Petri dish, so nothing is coming in through R_a. We have also shown the two different variable conductances representing the sodium and potassium channels. See Equation 6-2.

The hard part was to model the time and voltage dependence of the Na and K conductances. As you learned previously, their solution was to perform a series of voltage clamp experiments measuring both the total current and the current when the Na (Sodium) conductance was disabled. They did this with an injection probe and a recording probe, and a feedback circuit to insure that the injection probe applied just the right amount of current to hold the membrane potential at the desired value. They eliminated the sodium current by replacing the seawater with a solution containing Choline instead of Sodium chloride. Modern experiments use a puffer fish neurotoxin, Tetrodotoxin (TTX) to block the sodium channels.

By performing the experiments with different values of the clamp voltage, they were able to determine the time dependence and equilibrium value of the conductances at different voltages. Here Figure 6-13 is from one of their 1952 papers that shows the behavior of the K conductance when the voltage is stepped to 25 mV above the rest potential and then brought

back down to the rest potential. For different clamping voltages, they found different values of the maximum conductance, and different time behaviors.

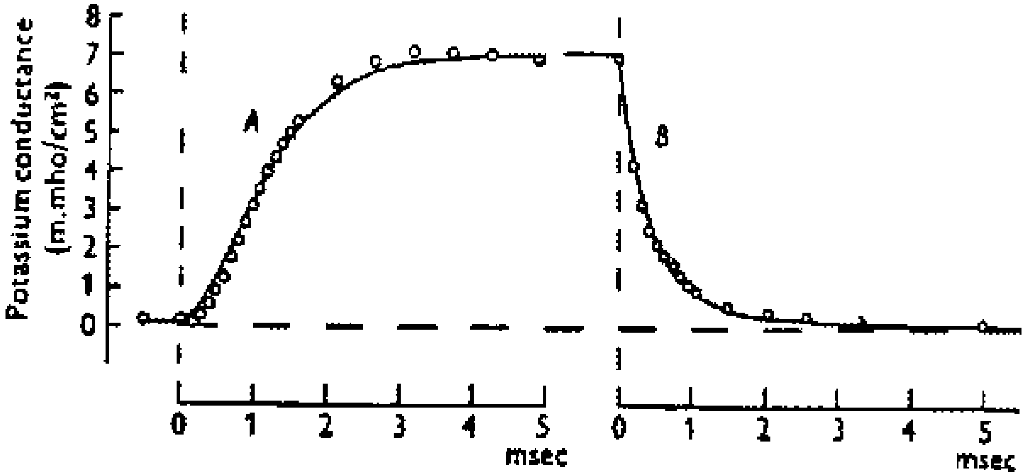


Figure 6-13. Clam Voltage for Different Values.

It would be tempting to fit the rise and fall of the conductance to an exponential function of time, but the experimental measurements were best, fitted by an exponential to the fourth power. They were able to fit the K conductance to an equation of the form as 6-3.

$$G_K = \bar{g}_K n^4 \qquad \text{Eq. 6-3}$$

Where n is called "activation variable" and has a simple exponential dependence governed by a single time constant, τ_n.

$$n(t) = n_\infty(V) - [n_\infty(V) - n_\infty(0)]e^{-t/\tau_n} \qquad \text{Eq. 6-4}$$

n_∞ is called the "steady state activation," i.e., the value reached by n when the membrane is held at a potential V for a long time. Hodgkin and Huxley were able to fit the voltage dependence of n_∞ and τ_n to an analytic function of voltage involving exponentials.

The tricky part is that when we are dealing with action potentials rather than a voltage clamp, n_∞ and τ_n are changing with the changing voltage, so we cannot use this equation for n. Instead, we have to use a differential equation that has this solution when V is constant,

$$\frac{dn(V)}{dt} = \frac{[n_\infty(V) - n(V)]}{\tau_n(V)} \qquad \text{Eq. 6-5}$$

This makes things pretty, hard if you are cranking out the numerical solution step-by-step on a calculator, but it is no big deal to solve on a computer. It is just one more, simple Differential Equation (DE) to solve numerically.

Their fit for the Na conductance was a little different, because they found that at a fixed voltage, the conductance rose with time and then decreased, so they had to fit it to a product

$$G_{Na} = \bar{g}_{Na} m^3 h \qquad \text{Eq. 6-6}$$

Here, m is the activation variable for Na, and h is called the "inactivation variable" since it becomes smaller when m becomes larger. The terminology is a little confusing though, because the conductance is large when the "inactivation" is large. m and h obey equations just like the ones for n, but their steady state values and time constants have different voltage dependences.

6.8.1. Further Details of the Hodgkin-Huxley Model

In their model, Hodgkin and Huxley represented the rate of change of the Potassium activation variable n as a rate equation for first-order kinetic process:

$$\frac{dn}{dt} = \alpha_n(V)(1-n) - \beta_n(V)n \qquad \text{Eq. 6-7}$$

The modern interpretation of this equation is that the opening of a K channel involves the motion of four physical gates between "permissive" and "non-permissive" states. If all gates are in a the permissive state, then the channel is open, and ions are allowed to flow. In a population of channels, n is the fraction of gates that are open. Thus, the conductance of these of channels is proportional to the fourth power of n. Here, α_n is the voltage-dependent rate constant for transitions to the permissive state, and β_n is the rate constant for transitions to the non-permissive state. By comparison with Equation 6-5, we can see that

$$\alpha_n(V) = \frac{1 - n_\infty(V)}{\tau_n(V)} \qquad \text{Eq. 6-8}$$

and

$$\beta_n(V) = \frac{1 - n_\infty(V)}{\tau_n(V)} \qquad \text{Eq. 6-9}$$

Likewise, the Na activation and inactivation variables obey the equations

$$\frac{dm}{dt} = \alpha_m(V)(1-m) - \beta_m(V)m \qquad \text{Eq. 6-10}$$

$$\frac{dh}{dt} = \alpha_h(V)(1-h) - \beta_h(V)h \qquad \text{Eq. 6-11}$$

with similar relationships between the rate constants and the activation variables and their time constants.

From their voltage clamp measurements of the activation variables and their time constants, Hodgkin and Huxley were able to make the empirical fits:

$$\alpha_n(V) = \frac{0.01(10-V)}{\exp\left(\frac{10-V}{10}\right)-1} \qquad \text{Eq. 6-12}$$

$$\beta_n(V) = 0.125\exp(-V/80) \qquad \text{Eq. 6-13}$$

$$\alpha_m(V) = \frac{0.1(25-V)}{\exp\left(\frac{25-V}{10}\right)-1} \qquad \text{Eq. 6-14}$$

$$\beta_m(V) = 4\exp(-V/18) \qquad \text{Eq. 6-15}$$

$$\alpha_h(V) = 0.07\exp(-V/20) \qquad \text{Eq. 6-16}$$

$$\alpha_m(V) = \frac{1}{\exp\left(\frac{30-V}{10}\right)+1} \qquad \text{Eq. 6-17}$$

Here voltages are measured with respect to a resting potential of zero (rather than the actual value of about -70 mV), and are given in mV. With these expressions for the rate constants, one can numerically solve Equations. 6-7, 6-10, and 11 for the time dependence of the activation variables, and use these in Equation 6-3 and Equation 6-6 to solve for the conductances. These can then be, used in Equation 6-2 to solve for the time behavior of the membrane potential.

As we will see in the next lecture on the Traub hippocampal pyramidal cell model, there are many other varieties of voltage-activated conductances found in neurons, in addition to those found in the squid giant axon. Most of these can be modeled using variations of the Hodgkin-Huxley model, with a similar notation, but with different values of the parameters and different values of the exponents for activation and inactivation. Some modelers use

equations similar to Equations 12-17, and others use expressions for the steady-state activation and time constant, which can be, related to these by Equations 6-8 and 6-9.

6.8.2. Summary of the Hodgkin-Huxley Model

The Hodgkin-Huxley model of the process by which action potentials are, generated in the giant axon of the squid lies at the basis of most neuronal models. Here is a brief summary of the equations and assumptions, which went into the model.

The mathematical model is, based upon the equivalent circuit for a patch of cell membrane. In the text of Principles of Neural Science 4th (fourth) Edition by Kandel, Eric, Schwartz, James, Jessell [15], this is Figure 9-14. The two variable conductances G_K and G_{Na} shown in the diagram represent the average effect of the binary gating of many Potassium and Sodium channels, and the constant "leakage conductance" G_L represents the effect of other channels (primarily chloride) which are always open. Each of these is associated with an equilibrium potential, represented by a battery in series with the conductance.

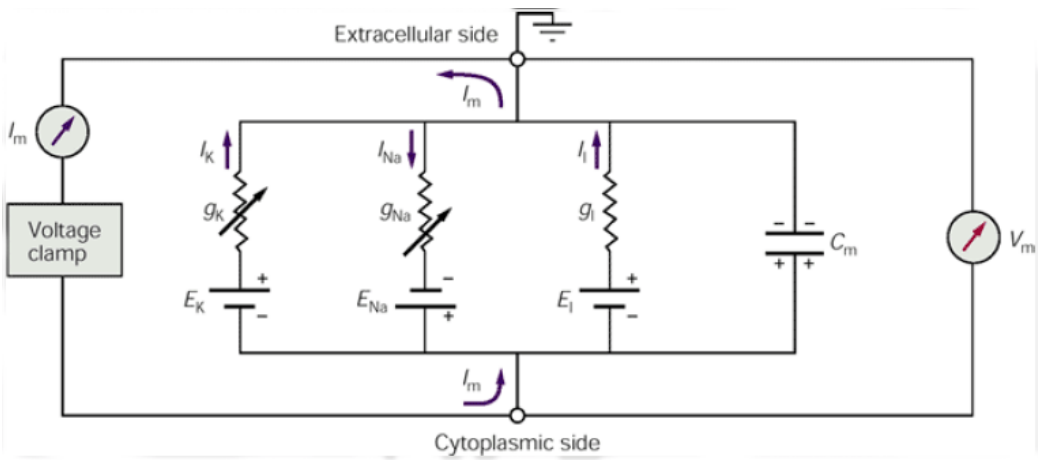


Figure 6-14. Electrical equivalent circuit of a nerve cell is being, held at a depolarized potential under voltage-clamp conditions [15].

In the Figure 6-14, the voltage-gated conductance pathways (G_K and G_{Na}) are represented by the symbol for variable conductance—a conductor (resistor) with an arrow through it.

The net current, which flows into the cell through these channels, has the effect of charging the membrane capacitance, giving the interior of the cell a membrane potential V_m relative to the exterior. From basic circuit theory, we know that the current, which charges a capacitor is equal to the capacitance times the rate of change of the voltage across the

capacitor. Ohm's law gives the current through each of the conductances, resulting in the equation

$$C_m \frac{dV_m}{dt} = G_{Na}(E_{Na} - V_m) + G_K(E_K - V_m) + I_{\text{inject}} \qquad \text{Eq. 6-18}$$

Here, an additional term I_{inject} has been, added to describe any currents, which are externally, applied during the course of an experiment. In principle, all that is, needed in order to find the time course of the membrane potential is to solve this simple differential equation.

The hard part was to model the time and voltage dependence of the Na and K conductances. As you can see from reading of Chapter 9 of reference 15, their solution was to perform a series of voltage clamp experiments measuring both the total current and the current when the Na conductance was disabled. This enabled them to calculate the K current and, from these currents and the known voltages, calculate the values of the two conductances. By performing the experiments with different values of the clamp voltage, they were able to determine the time dependence and equilibrium value of the conductances at different voltages. Figure 9-15 shows some typical results for the behavior of the K and Na conductances when the clamping voltage is stepped to several different values and then released. From these measurements, they were able to fit the K conductance to an equation of the form

$$G_K = \bar{g}_K n^4 \qquad \text{Eq. 6-19}$$

where n is called the "activation state variable" and has a simple exponential dependence governed by a single time constant, τ_n:

$$n(t) = n_\infty(V) - [n_\infty(V) - n_\infty(0)]e^{-t/\tau_n} \qquad \text{Eq. 6-20}$$

$n_\infty(V)$ is called the "steady state activation," i.e., the value reached by n when it is held at the potential V for a long period of time. Hodgkin and Huxley were able to fit the voltage dependence of n_∞ and τ_n to an analytic function of voltage involving exponentials. In the interest of brevity, we will not give these equations here. However, the plot of $n_\infty(V)$, shown further below, reveals that it is a monotonically increasing function of V, reaching a maximum value of 1.

In Figure 6-15, the increases and decreases in the Na+ and K+ conductances (G_{Na} and) shown here reflect the shifting of thousands of voltage-gated channels between the open and closed states.

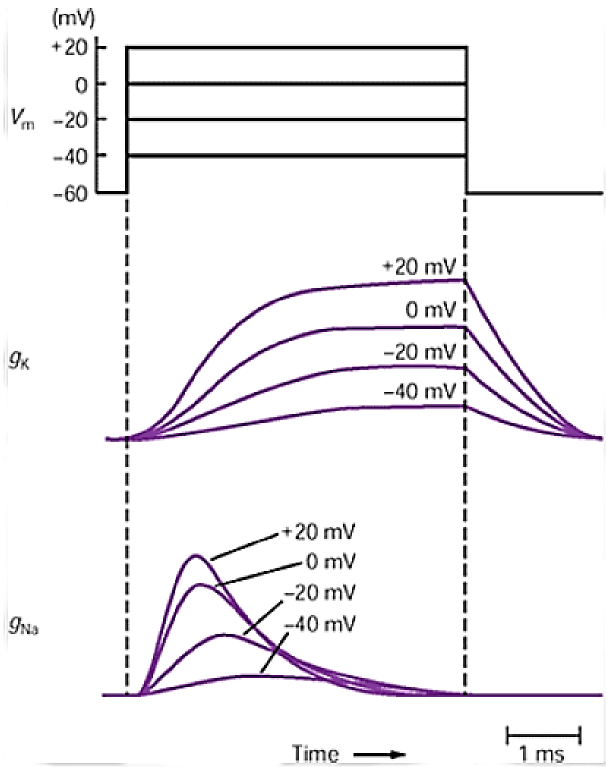


Figure 6-15. Voltage-clamp experiments show that Na+ channels turn on and off more rapidly than K+ channels over a wide range of membrane potentials [15].

If we are describing the time course of action potentials rather than the behavior during a voltage clamp, n_∞ and τ_n are changing along with the changing membrane potential, so we cannot use this equation for n. Instead, we use a differential equation, which has this solution when V is constant,

$$\frac{dn(V)}{dt} = \frac{[n_\infty(V) - n(V)]}{\tau_n(V)} \quad \text{Eq. 6-21}$$

Their fit for the Na conductance was a little different, because they found that at a fixed voltage, the conductance rose with time and then decreased (as shown in Figure 6-15), so they had to fit it to a product

$$G_{Na} = \bar{g}_{Na} m^3 h \quad \text{Eq. 6-22}$$

Here, m is the activation variable for Na, and h is called the "inactivation state variable," since it becomes smaller when m (and the membrane potential) becomes larger. m and h obey equations just like the ones for n, but with different voltage dependences for their steady state values and time constants. These voltage dependences are, shown in the

following plot, derived from Hodgkin and Huxley's fit to their experimental results. See Figure 6-16.

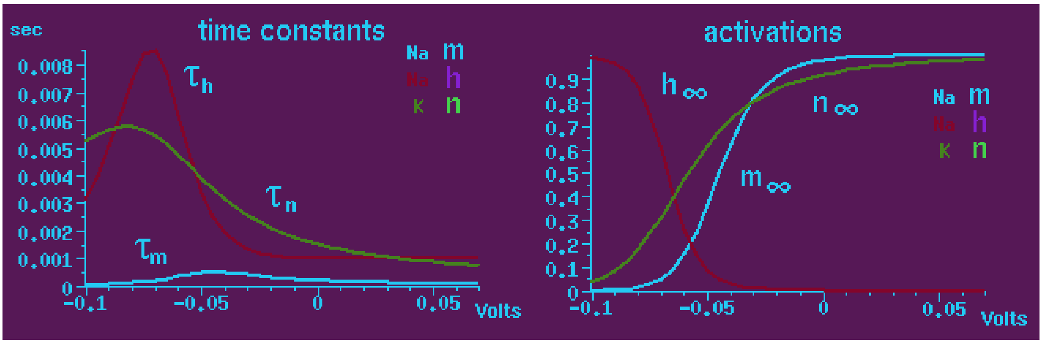


Figure 6-16. Illustration of Hodgkin and Huxley's from Experimental Result.

We now have all that was, needed by Hodgkin and Huxley to reconstruct the action potential. For a given injection current I_{inject}, Equation 6-18 is solved for V_m, using Equations 6-19 and 6-22 for the conductances. These two equations must be, solved simultaneously with Equation 6-21 and the two analogous equations for m and h. These last equations make use of the voltage dependent quantities shown in the plot of Figure 6-16.

This plot shows that although the time constants vary with voltage, the time constant for the Na activation variable m is about an order of magnitude less than that for the Na inactivation and the K activation throughout the entire range. This means that during an action potential, when the voltage is high and m is large, and h is supposed to be small, it will take a while for h to decrease. In addition, it will take n a while to become large and contribute to the opposing K current.

As we will see in a later lecture, the behavior of these quantities is the key to understanding the time course of the action potential, as well as the phenomenon of the "refractory period" following the action potential.

6.9. Modeling the Brain: Simplified versus Realistic Models

This is the beginning of the "and Computers" part of the "Brains, Minds, and Computers" course. G_K If I had to choose one phrase to describe the topics we will cover during the next month, we would pick "machine intelligence" - simulating *Brain* or *Mind* with a computer.

So far, the course has been all about Brains, on the level of basic neurobiology. This is for a good reason. If our goal is to understand how brains work, perhaps with the hope of making "smarter computers," we need to understand the components that form the "wetware" of the brain. This is essentially a bottom-up reductionist approach. By starting by trying to understand the properties of parts of neurons -- cell membranes, ion channels, and synapses, we hope to gain an understanding of single neuron behavior, and then network behavior, and finally the behavior of the entire organism.

This has an analogy with understanding how the hardware of a computer works. An understanding of atoms leads to an understanding of silicon semiconductor junctions, which leads to an understanding of transistors, and then the Very Large Scale Integrated (VLSI) circuits used to construct a computer. Can we do the same with brains? Alternatively, should we start with a more high-level approach?

What about Mind? If we understand the Brain, will we then understand the Mind? The (unproven) assumption that we take in this course is that "Mind" somehow arises out of the function of Brain. i.e., those mental states should somehow be, related to patterns of firing of neurons and their synaptic connections.

We will not say much more about Mind in this course, so we would like to spend a little time to at least, give names to some of the subjects that relate Brains, Minds, and Computers. We like to picture the relationships with a figure like this, where the direction of the arrows indicates how one area of study influences another:

Cognitive Neuroscience sheds light on the relationship $B \rightarrow M$ by asking "what does the functioning of the brain have to do with states of mind?." It explores the other direction, $B \leftarrow M$ by asking, "What do subjective states of mind imply about the functioning of the brain?" In general, it attempts to understand mind in terms of what we know about the biology and physiology of the brain. It places a lot of emphasis on trying to understand the internal representation of external events.

Artificial Intelligence (AI) implements high-level abstract models of human cognition (Mind) to make computers more "intelligent."

Cognitive Science uses computer models (among other approaches) to understand the functioning of Mind.

Artificial Neural Networks (ANN) use knowledge of the behavior of neurons to make computer algorithms (or hardware implementations) that are more "intelligent" than conventional computers.

Computational Neuroscience is usually, understood to mean the use of computer models to help understand the way that neurons or networks of neurons process information, or "compute." However, there is often disagreement as to the level of detail and biological realism that is needed in the models in order to achieve this goal.

There are a number of techniques and tools of experimental neuroscience that have provided the necessary data to construct quantitative theoretical models in cognitive neuroscience and computational neuroscience:

1) Intra- and extracellular recording **-** Simultaneous multicellular recording is analogous to using a logic analyzer. This makes it possible to monitor the activity of neurons while animals are engaging in various behaviors. Firing patterns in specific regions can be associated with particular mental processes
2) The results of cortical lesions on behavior shed light on the role of particular brain regions. (Much has been, learned from the cognitive defects of patients who have experienced lesions from injury or stroke. The results of surgery on animal subjects is also a source of information)
3) Non-invasive imaging techniques - Positron Emission Tomography (PET), Magnetic Resonance Imaging (MRI), Electroencephalography (EEG), Magnetoencephalograpy (MEG), a recent counterpart to EEG which measures magnetic fields due to ionic

currents in the brain), and voltage sensitive dyes, can map activity in cortical areas to mental processes and behavior. These have comparatively low resolution, but can help us create a "block diagram"

4) Neuron staining techniques - The technique goes back to the pioneering work of Ramon y Cajal in the late 19th century. We can identify components and connections, to some extent, but we may have to make shrewd guesses and look for ways to test them
5) Confocal microscopy gives us a 3-D view of individual neurons
6) Radioactive tracers injected into neurons can map axonal connections.
7) Patch clamp techniques allow recording from individual ion channels
8) Voltage sensitive dyes change color according to the membrane potential. Other dyes are sensitive to the intracellular concentration of calcium ions.

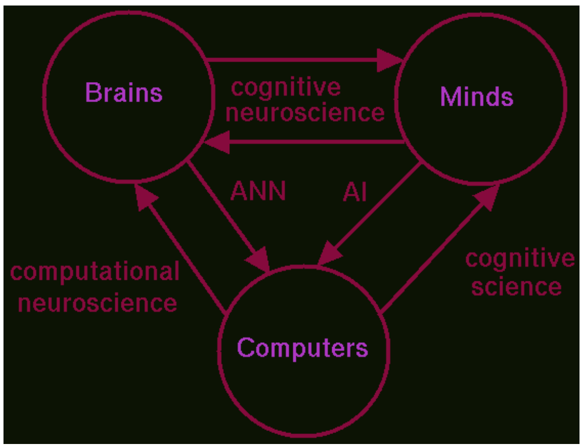


Figure 6-17. Disciplines that Relate Brains, Minds, and Computers.

6.10. USE OF COMPUTERS TO MODEL SOME ASPECT OF "BRAIN" OR "MIND"

There are two main motivations for making computer models of "*Brain*" or "*Mind.*" Pure Science tries to understand "how the brain computes" or "how intelligence works" by using computer simulations to test and refine their theoretical models. The Practical Engineering approach is concerned with learning how to make "*better*" or "more *intelligent*" computers.

Clearly, there is a lot of overlap between these two goals -- an engineer needs to know something about the workings of the brain in order to apply this knowledge to computers. Of course, a philosopher will want to start by answering the question "what is intelligence?" This can be a lot of fun to debate with friends late at night after a few beers, but we are not going to say much about it because it could take forever. We all recognize that although computers are very fast, there are many things, which people, and even very primitive animals, do better and faster. For example, which of these is not a tree? See Figure 6-18.

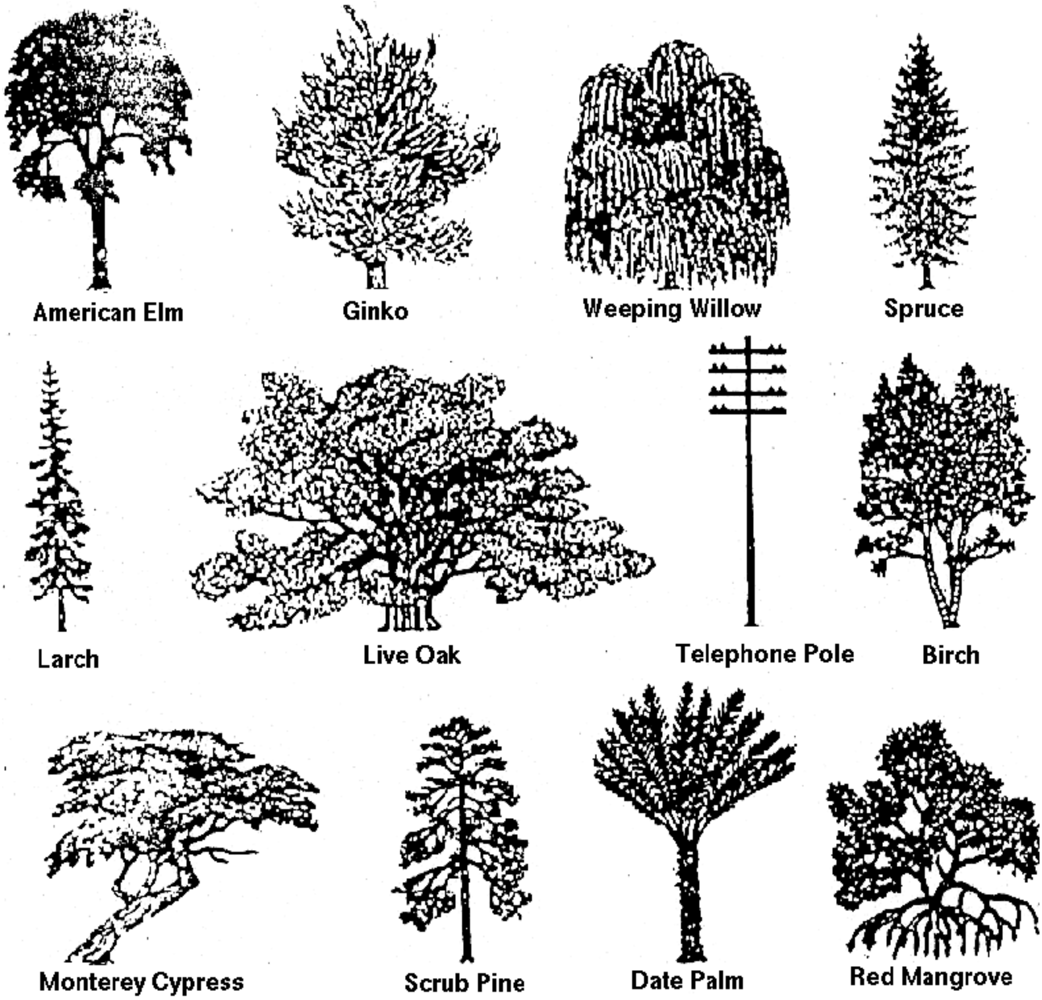


Figure 6-18. Object and Pattern Recognition.

On the other hand, for that matter, what does Figure 6-19, represent? Would a computer program that was, trained to recognize photographs of cats be able to recognize it as a cat?

These are hard problems for a computer to solve, even though we can solve them easily. All three approaches (Computational Neuroscience, AI, and Artificial Neural Nets) attempt to understand how problems like these can be, solved. It is interesting to think about what we do when we solve them. What goes on in your mind or in your brain when you solve one of these problems?

This introspective approach is, basically the one taken by traditional artificial intelligence (AI). How would you tell someone what you did to solve these problems? Or is your answer just a rationalization of something you did much less consciously?

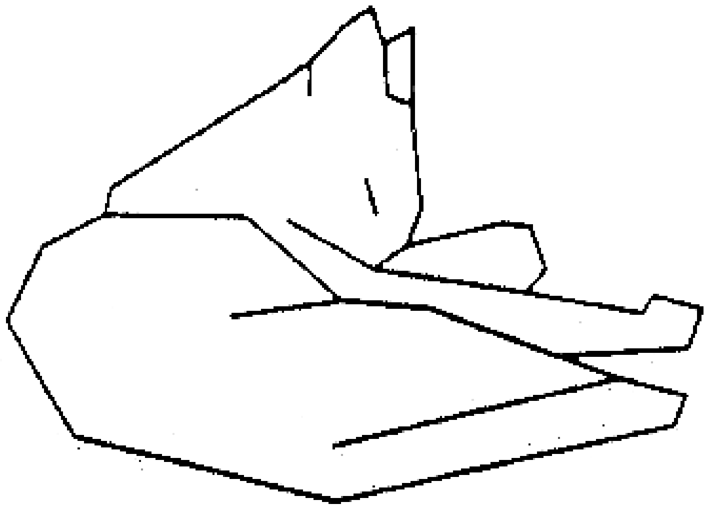


Figure 6-19. Lying Down Cat (Maybe).

Computational Neuroscience has mostly the goals of pure science, although it may shed some light on how to accomplish the goal of practical applications. The other two approaches, AI and Neural Networks, both offer a more practical approach toward implementing "machine intelligence." In order to understand how the brain works, we might have to model it down to the detail of single ion channels, but (at least with present computer technology) we would not expect our simulations to be a practical way of performing "brain-like" computations. We need to leave out some details.

So, how do we make a computer "intelligent"?

This knotty problem is like a tangled ball of wire as it can be, seen in Figure 6-13x, which we can attack from two sides. The two different practical approaches are "Traditional AI" and "Artificial Neural Nets." The first approach has its roots in psychology, and might be, called "Minds and Computers." It tends to focus on high-level abstractions like "the mind." The other approach, which I'll call "Brains and Computers," tries to apply what biology and physiology tell about the way the brain works.

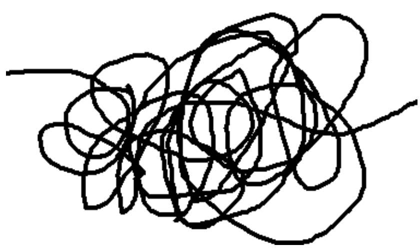
"Minds and Computers" • Artificial Intelligence • Psychology (behavior) • High level, top-down • Macroscopic approach		**"Brains and Computers"** • Artificial Neural Nets or Realistic Models • Biology (physiology) • Low level, bottom-up • Microscopic approach

Figure 6-20. Illustration of Tangled Ball.

Traditional Artificial Intelligence falls into both categories of goals. It represents the opposite extreme in terms of the level of microscopic detail in the models. The models tend to be high-level abstract models based more on psychology or linguistics, rather than on biology. (Example: Freudian psychology uses non-biological concepts like the ego, superego and id. Jungian psychology uses others. Even if there are no biological structures corresponding to these functions, they may be useful models for understanding mental processes.) Artificial Intelligence (AI) has been, used as a tool for understanding “how intelligence works.” However, we think it is most useful as a practical tool for making computers more like minds.

Artificial Neural Networks falls mainly in the category of practical applications (although many workers in this field believe that simplified models can shed light on the workings of the brain). Here, the idea is to perform computations with networks of neuron-like elements. The approach has a lot in common with computational neurobiology, but we would like to leave out as many of the complicated biological details as we can safely ignore.

People with different interests and backgrounds tend to have different opinions/prejudices about the best end of the problem to attack. Having worked as a solid-state physicist, I like the bottom-up approach. If I want to understand the behavior of fluids, we start with a computer simulation of interacting molecules and try to understand their macroscopic behavior based on what we know about the microscopic behavior of the components. On the other hand, someone who needs to predict the behavior of 40 weight racing oil will not find this approach very helpful, at least in the short term. He or she may need a more empirical approach, using many “black boxes” which are not understood in detail. In addition, we will see that some types of “intelligent behavior” are better, treated with one approach than the other. You might think about which approach seems best for the two pattern recognition problems, which we posed. Do you take a cognitive approach (thinking about how you would describe the differences between a tree and a telephone pole)? Or is your process less conscious? Or do you think about how a frog recognizes a fly? A frog probably does not intellectualize things much. A pattern falls across its retina, its tongue goes out, and ZOT!

The nice thing about being an engineer is that you get to pick, and choose between the various alternatives, and choose the one that seems to offer the best possibilities at the time. Sometimes when untangling a ball of wire, you work on one loose end until progress slows down and then switch to the other. You hope to eventually, meet in the middle. Our own opinion is that recently, the approach of trying to understand the biological details of brain function has been more fruitful than high-level models of cognition.

6.11. Overview of Artificial Neural Networks

This is a very short summary of the Artificial Neural Network (ANN) approach to neural modeling. Our goal here is to explain the neurobiological basis of ANN models by identifying the characteristics of biological neurons and neural networks that should go into a simple model of what we might call “biologically-inspired computation.” We cannot model the brain down to every level of detail, so we have to decide what features are important for the types of computation, which are performed by the brain. With any sort of modeling of a complicated system, this question always comes up.

Someone once said that if we really wanted biological realism, then airplanes should have feathers. Then, there is the story of the physicist who was going to model a racehorse to predict the winner of a horse race. (“First, let us assume a spherical horse ...”) We do not want our model to be either an airplane with feathers or a spherical horse. The decision to leave out certain details should be an informed one, not based on ignorance of the details.

6.12. PROPERTIES OF “WET” NEURONS

What do we know about the behavior of real neurons, which we would like to include in our model?

1. Neurons are integrating (summing) devices. Each input spike causes a Postsynaptic Potential (PSP) caused by current pulse, which adds to the charge on the membrane capacitance, gradually increasing the membrane potential until the potential in the axon hillock reaches the threshold for firing an action potential.

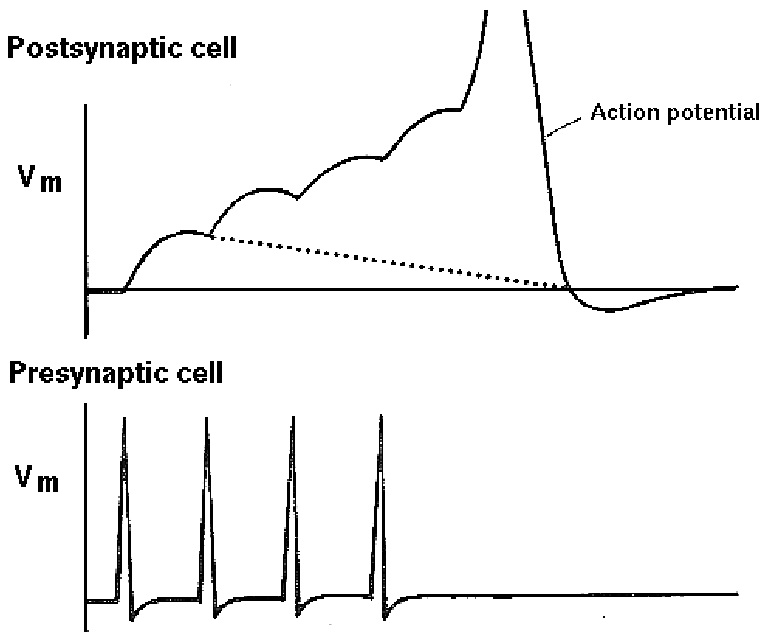


Figure 6-21. Postsynaptic Potential Illustration.

2. Neurons receive both excitatory and inhibitory inputs, with differing and modifiable synaptic weights. This weighting of synaptic inputs can be, accomplished in a variety of ways. The efficiency of a single synaptic connection depends on the surface area and geometry of the synapse and the number of vesicles containing neurotransmitter on the presynaptic side. A neuron, which interacts strongly with another, may make many synapses with the same target neuron, so we can use multiple synapses to increase the weight of a connection between two neurons.

3. Neurons have a non-linear input/output relationship

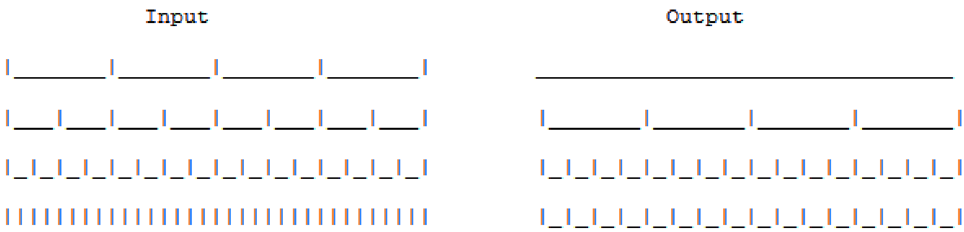


Figure 6-22. Nonlinear Input and Output Relationship.

Remember from electrical engineering course, if we plot the input firing rate versus the output-firing rate, we generally get something like this, with a threshold, and a saturation level arising from the refractory period for firing.

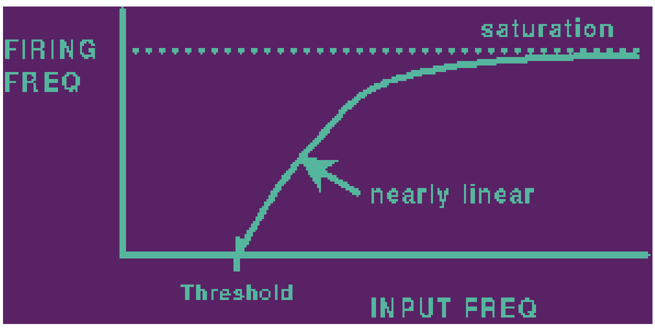


Figure 6-23. Input Firing Rate vs. the Output Firing.

6.13. The Generic Model "Neurons"

There are many variations on the models, which have been, proposed for artificial neuron-like computational units. Other lectures in this course will cover some of them in detail and a few of them superficially. Most of them have these basic features.

1. The output of the i th neuron is represented by a voltage level, V_i, instead of a sequence of spikes. Sometimes this is a continuously variable analog voltage, which represents the firing rate of a real neuron. In other models, the output is a binary value, representing either firing or not firing.

$$u_i = \sum_j W_{ij} V_i + I_i \quad \text{Eq. 6-23}$$

2. The input to the i th neuron is a weighted sum of the outputs of the neurons that make connections to it, plus any external inputs. The connection weights, W_{ij}, may be positive or negative, representing excitatory or inhibitory inputs.

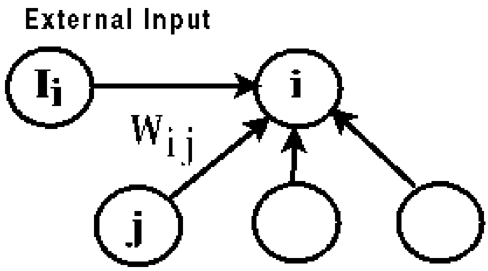


Figure 6-24. Neuron Connection Weighted Sum.

3. The output is related to the input by some non-linear function $V_i = f(u_i)$, which may look like:

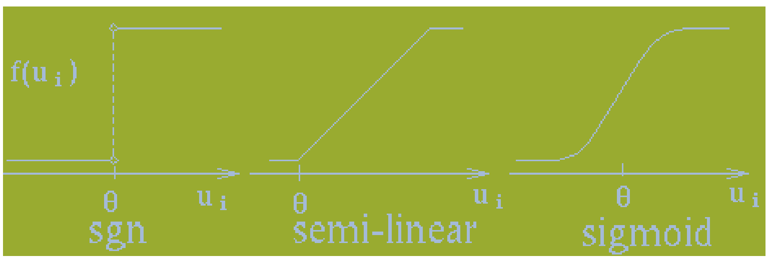


Figure 6-25. Depiction Function $V_i = f(u_i)$.

The function often has some sort of threshold parameter (theta) that allows different neurons to have different thresholds for activation. We will see different variations on this basic paradigm. Almost all of them have in common the idea that the so-called "neurons" receive inputs from the outside world, or from other neurons, which are, multiplied by some weighting factor and summed. The output or "activation" is, formed by passing the input through some sort of "squashing" function. This output often becomes the input to other neurons.

The situation we are describing is something like this with some units ("neurons") receiving external inputs (I), some presenting an external output (O), and others being intermediate (hidden) units that only connect with other units. The output of one unit is passed on to become part of the weighted input of another, finally reaching the output units. The states of the neurons may be updated in parallel (synchronously), or one at a time (asynchronously). In most neural network models, the network is, designed so that the outputs of all the neurons will eventually settle down to a steady state when the external input is held constant. It may take a number of iterations of the update process for this to occur. (Biological networks are different in this respect - their output is usually continuously changing.) Various

learning algorithms such as "backpropagation" are used to modify the weights in order train the network to perform some mapping between inputs and outputs.

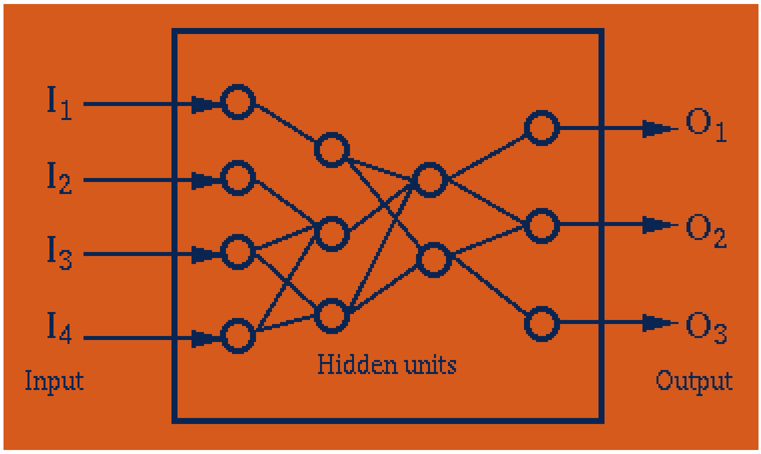


Figure 6-26. Sequence of Input and Output Process.

A question to discuss: What possibly significant features have we left out of our model? Are there any important ways in which it is different from real neurons?

Spatial effects - What are the consequences of having inputs to various parts of an extended dendritic tree? Can the right weights in a ANN model take this into account? Perhaps not. For example, shunting inhibition by the activation of channels close to the soma can "gate" excitation from more distant regions. The various ways that that computations are performed in the dendritic tree is a current research topic.

Temporal effects - is firing rate everything? Our artificial neurons have an output which either corresponds to a firing rate or simply to "firing" or "not firing." However, the pattern of firing of a biological neuron can encode information. The phase relationship and correlations between the firing of neurons can be important. If a neuron receives input from several others that are firing in unison, the effect will be larger than if their firings were uncorrelated. There may be even more subtle ways that the spacing of action potentials can convey information. Consider the differences between the firing patterns of neurons **A**, **B**, and **C**, which all have the same average firing rate:

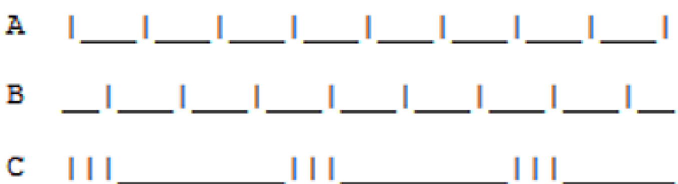


Figure 6-27. Temporal Effects for Firing Patterns of Neurons.

If another neuron receives synaptic inputs from neurons **A**, **B**, or **C**, can you see how the results might vary?

Here are some specific examples:

- As little as three spikes from the retinal system of the blowfly can tell it which way to turn. Obviously, many bits of information would be required for any sort of precision. This implies that analog information conveyed by spike timing is being used
- The barn owl uses phase differences and delays for sound localization
- In the cat visual system, spike timing is, preserved through four layers of the visual cortex - there must be a reason for this.
- Spiral ganglion cells in the cochlea phase lock when stimulated by pure tones

(Some others are listed in L. Watts, Advances in Neural Information Processing Systems 6 pp. 927-934 (1994)).

6.14. REALISTIC NEURAL MODELING

The following lectures will emphasize biologically realistic neural models. However, there are many reasons why one should NOT make realistic neural simulations. We should at least be aware of them. The usual excuses are:

- The human brain is large and complex, with
  - about 20 x 10^9 neocortical neurons
  - and 200 x 10^{12} synapses
  - There are about 100 x 10^9 neurons in the CNS
  - Neurons have specialized and complex dendritic structures
  - A cortical neuron has many types of ionic conductances
  - The “wiring diagram” is highly specialized and detailed
- Computers are too slow (but the 1989 GENESIS olfactory cortex simulation had 4500 neurons and ran on a 0.02 GHz Sun 3)
- The usual “solutions”:
  - Artificial neural networks (e.g., feed-forward with backpropagation)
  - Simple networks of integrate-and-fire “neurons”

If we plunge ahead and strive for maximal biological realism, here are some of the challenges that we face:

There are large varieties of neurons that have a shape (morphology) and behavior that are specific to their function in the network. Here are some types of neurons found in monkey visual cortex;

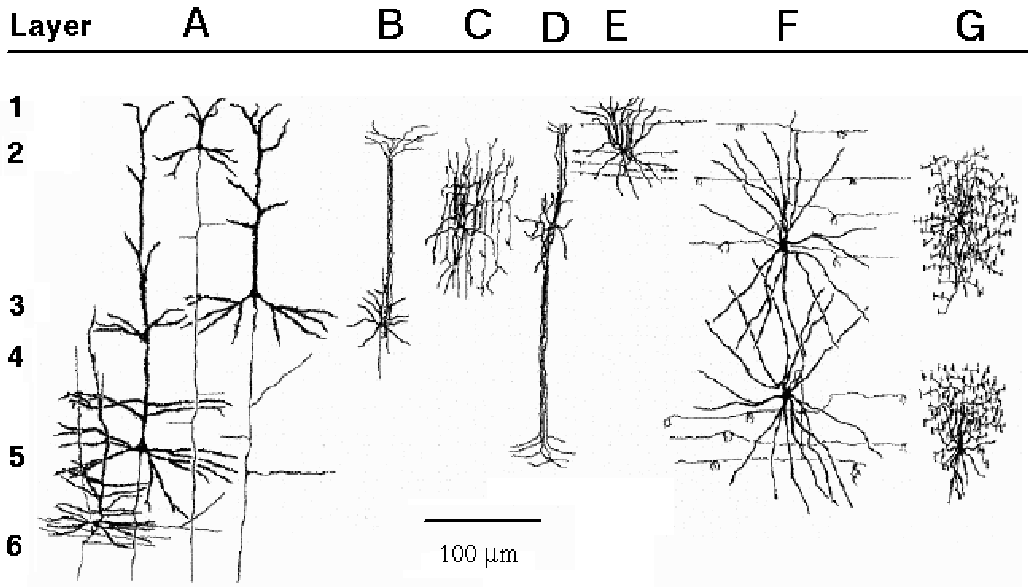


Figure 6-28. (A) Pyramidal, (B) Spiny stellate, (C) Bi-tufted, (D) Double bouquet, (E) Small basket, (F) Large basket, (G) Chandelier.

There are many varieties of spiking patterns in cortical neurons

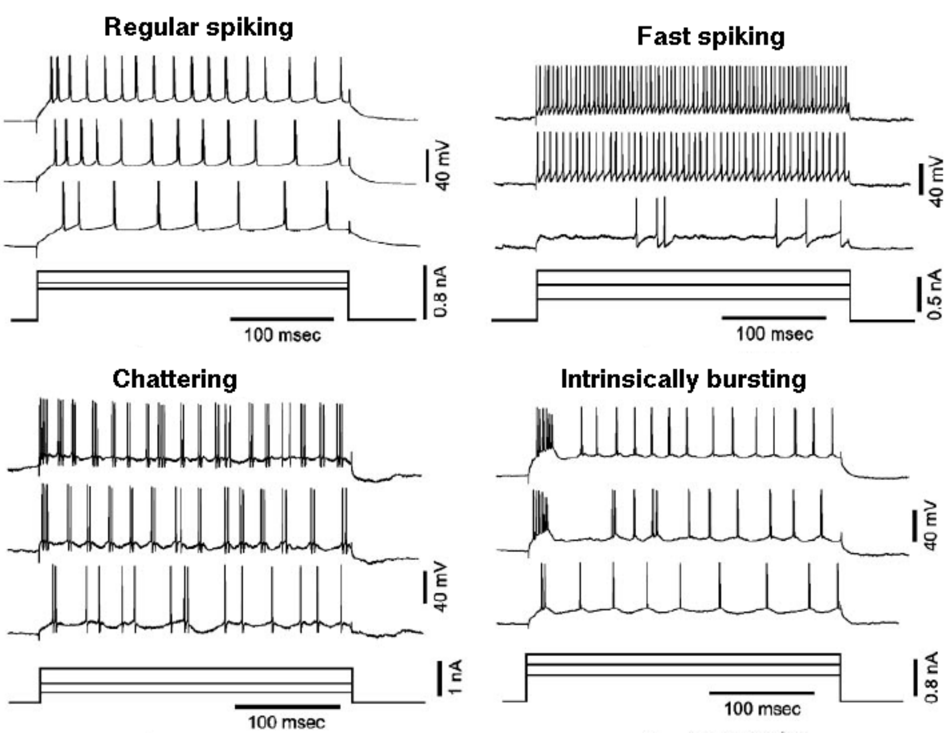
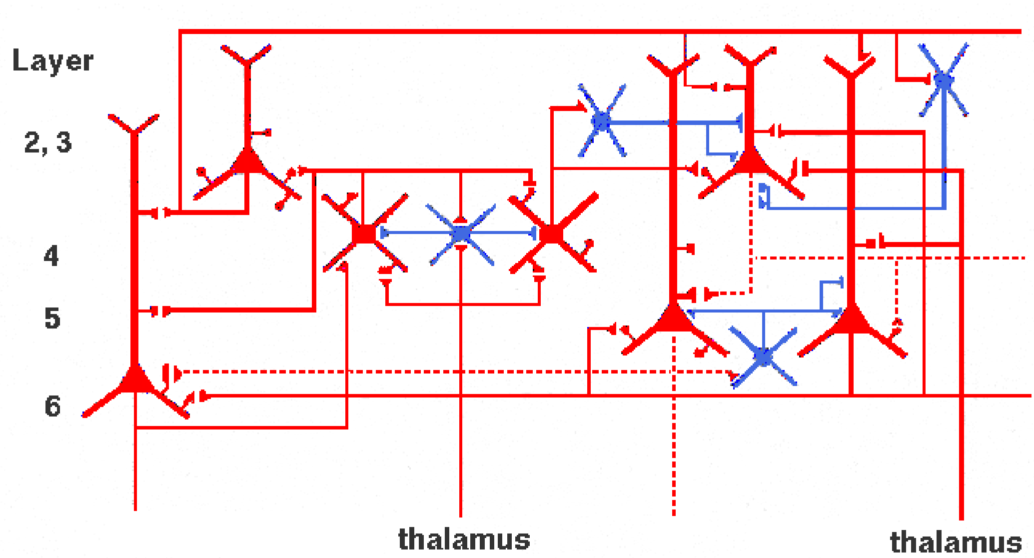


Figure 6-29. Illustration of Various Spiking Pattern in Cortical Neurons.

Even a "simplified" neocortical wiring diagram, such as this one is very complex with detailed and specific patterns of connections:



Based on: G. M. Shepherd, "Synaptic Organization of the Brain," 5th edition (2004).

Figure 6-30: Spiny neurons and connections are red; GABAergic are blue.

Nevertheless, there are also some good reasons to face this challenge:

- It is better to put the details in first, then and then simplify after you understand which details are important
- Bad simple models can often fit experiment well, and are not a good way to test a hypothesis.
- There is a symbiotic relationship between modeling and experiments
- Hard problems need many different approaches
- Many good single cell models are available to use in networks
- Computers are getting faster and more parallel.

6.15. Some Specific Models of Artificial Neural Nets

In the last lecture, I gave an overview of the features common to most neural network models. Figure 6-28 is illustrating more details that, you can see a diagram summarizing the way that the net input u to a neuron is, formed from any external inputs, plus the weighted output V from other neurons. This is used to form an output $V = f(u)$, by one of various input/output relationships (step function, sigmoid, etc.). These usually involve a threshold parameter, theta. At the bottom of the figure, there is a typical network, with input units receiving external inputs, hidden units, which communicate only with other neurons, and output units whose outputs are visible to the outside world.

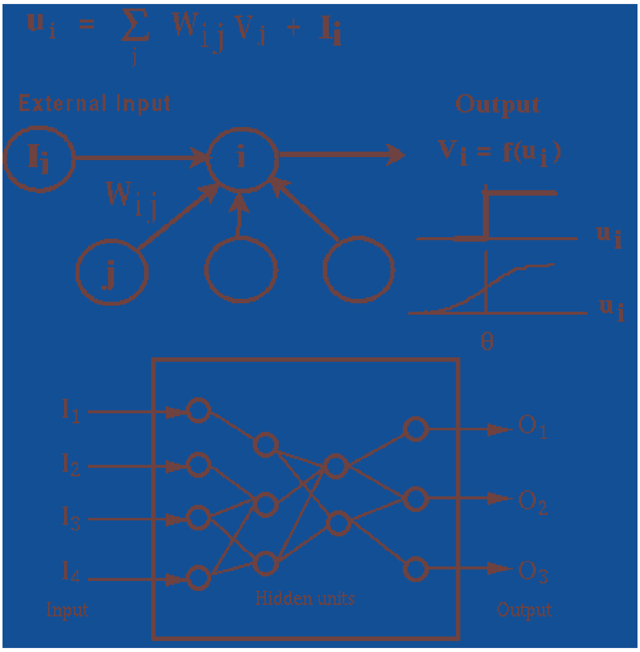


Figure 6-31. Summarization of Net Input to a Neuron from Net External.

6.16. McCullogh-Pitts Model

In 1943, two electrical engineers, Warren McCullogh and Walter Pitts, published the first paper describing what we would call a neural network. Their “neurons” operated under the following assumptions:

1) They are binary devices ($V_i = [0,1]$)
2) Each neuron has a fixed threshold, theta
3) The neuron receives inputs from excitatory synapses, all having identical weights. (However it may receive multiple inputs from the same source, so the excitatory weights are effectively positive integers).
4) Inhibitory inputs have an absolute veto power over any excitatory inputs.
5) At each time step the neurons are simultaneously (synchronously) updated by summing the weighted excitatory inputs and setting the output (V_i) to 1 if the sum is greater than or equal to the threshold AND if the neuron receives no inhibitory input.

We can summarize these rules with the McCullough-Pitts output rule

$$V_i = \begin{cases} 1 \ : \ \sum_j WV_j \geq \theta \ \text{ AND no inhibition} \\ 0 \ : \ \text{otherwise} \end{cases} \qquad \text{Eq. 6-24}$$

and the diagram associated with Equation 6-18 is depicted as Figure 6-29 here.

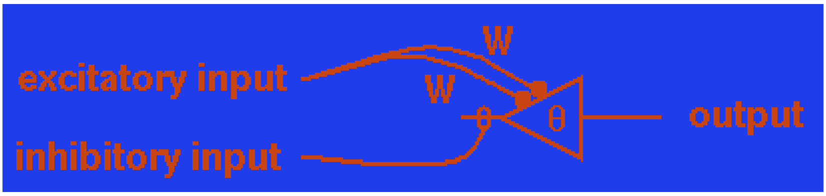


Figure 6-32. Illustration of Equation 6-18.

Using this scheme, we can figure out how to implement any Boolean logic function. As you probably know, with a NOT function and either an OR or an AND, you can build up XOR's, adders, shift registers, and anything you need to perform computation.

We represent the output for various inputs as a truth table, where 0 = FALSE, and 1 = TRUE. You should verify that when W = 1 and θ = 1, we get the truth table for the logical NOT,

Table 6-1. Truth Table

V_{in}	V_{out}
1	0
0	1

By using the circuit shown in Figure 6-30, we can have;

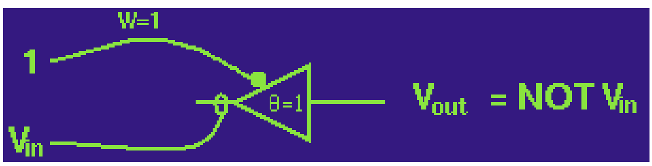


Figure 6-33. Logical Circuit with NOT.

With two inputs V_1 and V_2, and W =1, we can get either an OR or an AND, depending on the value of theta as presented in Equation 6-25:

$$\begin{cases} \text{If} \quad \theta = 1 \Rightarrow V_{\text{out}} = V_1 \;\; \text{OR} \;\; V_2 \\ \text{If} \quad \theta = 2 \Rightarrow V_{\text{out}} = V_1 \;\; \text{AND} \;\; V_2 \end{cases} \qquad \text{Eq. 6-25}$$

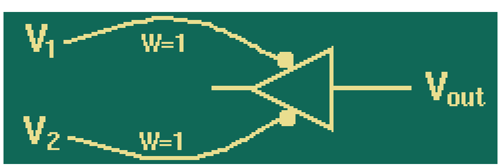


Figure 6-34. Logical Circuit with an OR or an AND.

As exercise, can you verify that with these weights and thresholds, the various possible inputs for V_1 and V_2 result in this table?

Table 6-2. Exercise Table

V_1	V_2	OR	AND
0 0 1 1	0 1 0 1	0 1 1 1	0 0 0 1

The exclusive OR (XOR) has the truth table

Table 6-3. The Exclusive OR (XOR) Truth Table

V_1	V_2	XOR
0 0 1 1	0 1 0 1	0 1 1 0

It cannot be represented with a single neuron, but the relationship XOR = (V_1 OR V_2) AND NOT (V_1 AND V_2) suggests that it can be represented with the network.

As part of the above exercise, explain to your own satisfaction that this generates the correct output for the four combinations of inputs. What computation is being, made by each of the three "neurons"?

These results were very encouraging, but these networks displayed no learning. They were essentially "hard-wired" logic devices. One had to figure out the weights and connect-up the neurons in the appropriate manner to perform the desired computation. Thus, there is no real advantage over any conventional digital logic circuit. Their main importance was that they showed that networks of simple neuron-like elements could compute.

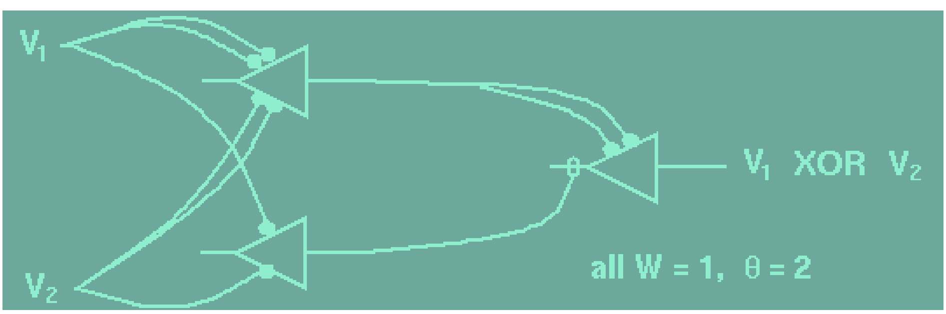


Figure 6-35. The Exclusive OR (XOR) Network.

6.17. THE PERCEPTRON

The next major advance was the perceptron, introduced by Frank Rosenblatt in his 1958 paper. The perceptron had the following differences from the McCullough-Pitts neuron:

1) The weights and thresholds were not all identical.
2) Weights can be positive or negative.
3) There is no absolute inhibitory synapse.
4) Although the neurons were still two-state, the output function f(u) goes from [-1,1], not [0,1]. (This is no big deal, as a suitable change in the threshold lets you transform from one convention to the other).
5) Most importantly, there was a learning rule.

Describing this in a slightly more modern and conventional notation (and with $V_i = [0,1]$) we could describe the perceptron like the one in Figure 6-33

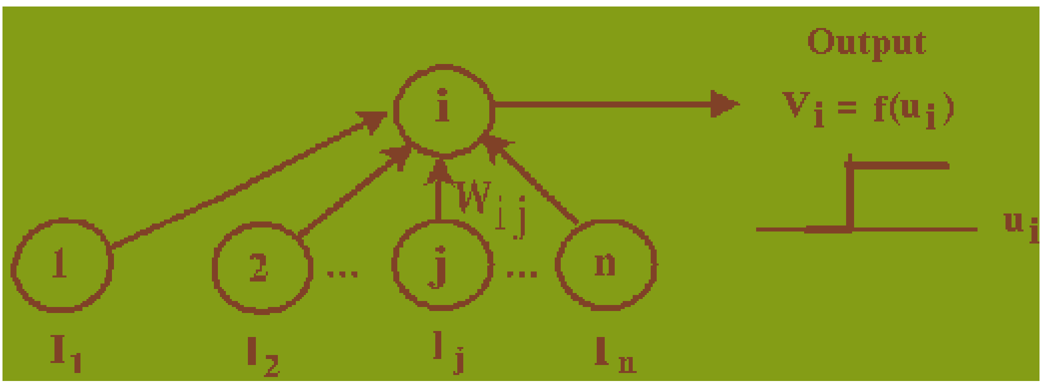


Figure 6-36. The Perception Diagram.

This shows a perceptron unit, i , receiving various inputs I_j , weighted by a "synaptic weight" W_{ij} .

The i th perceptron receives its input from n input units, which do nothing but pass on the input from the outside world. The output of the perceptron is a step function:

$$V_i = f(u_i) = \begin{cases} 0 & : \quad u_i < 0 \\ 1 & : \quad u_i \geq 0 \end{cases} \qquad \text{Eq. 6-26}$$

and

$$u_i = \sum_j W_{ij} V_j + \theta_i \qquad \text{Eq. 6-27}$$

For the input units, $V_i = I_j$. There are various ways of implementing the threshold, or bias, θ_i. Sometimes it is subtracted, instead of added to the input u, and sometimes it is included in the definition of $f(u)$.

A network of two perceptrons with three inputs would look like:

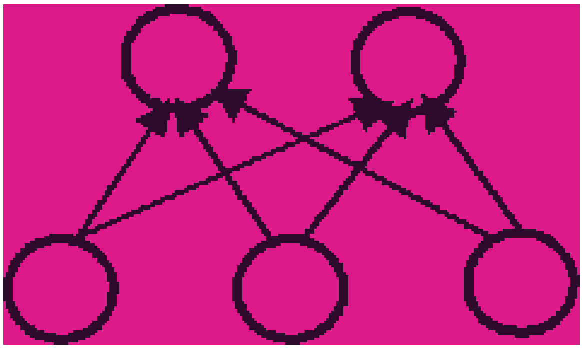


Figure 6-37. Depiction of Two Perceptrons.

Note that they do not interact with each other - they receive inputs only from the outside. We call this a “single layer perceptron network” because the input units don't really count. They exist just to provide an output that is equal to the external input to the net.

The learning scheme is very simple. Let t_i be the desired “target” output for a given input pattern, and V_i be the actual output. The error (called “delta”) is the difference between the desired and the actual output, and the change in the weight is, chosen to be proportional to delta. Specifically, $\delta_i = (t_i - V_i)$ and $\Delta W_{ij} = \varepsilon \delta_i V_j$, where is the *learning rate*.

Can you see why this is reasonable? Note that if the output of the i th neuron is too small, the weights of all its inputs are, changed to increase its total input. Likewise, if the output is too large, the weights are, changed to decrease the total input. We will better understand the details of why this works when we take up back propagation. First, an example can be, shown that:

6.17.1. Perceptron Learning of OR (by Pattern)

Before we can start, we have to ask, “How can we use this rule to modify the threshold or bias term, theta?”

Answer: treat theta as the weight from an additional input which is always “on” (V = 1). Now, consider the net:

Figure 6-38. Diagram of Net Example.

Unit 3 (the perceptron) receives inputs from the two input units 1 and 2, weighted by W_{31} and W_{32}, and a constant input of 1, weighted by theta3.

Let $\varepsilon = 0.5$ and initially set all the weights to $W_{31} = W_{32} = \theta_3 = 0$.

Then, we have

$$u_3 = W_{31}V_1 + W_{32}V_2 + \theta_3 \times 1$$
$$V_3 = \begin{cases} 0 & : \quad u_i < 0 \\ 1 & : \quad u_i \ge 0 \end{cases} \qquad \text{Eq. 6-28}$$

The error terms is $\delta_3 = t_3 - V_3$. This means that the change in weight will be $\Delta W_{3j} = 0.5\delta_3 V_j$, and the change in the bias is $\Delta\theta_3 = 0.5\delta_3 \times 1$.

Now fill in this table showing the results of each, iteration stopping when there is no further change through the presentation of all four patterns. We call each set of four patterns an “epoch.” In this case, we are “training by pattern” because we adjust the weights after each pattern. Sometimes, nets are “trained by epoch,” with the net change in weights applied after each epoch. (I'll do the first iteration).

How many epochs does it take until the perceptron has been trained to generate the correct truth table for an OR? Note that, except for a scale factor, this is the same result, which McCullogh and Pitts deduced for the weights and bias without letting the net do the learning. (Do you see why a positive threshold for a M-P neuron is equivalent to adding a negative bias term in the expression for the perceptron total input u?)

Table 6-4. Example Table

V_1	V_2	t_3	u_3	V_3	δ_3	New W_{31}	New W_{32}	New θ_3
0 0 1 1	0 1 0 1	0 1 1 1	0	1	-1	0	0	-0.5
0 0 1 1	0 1 0 1	0 1 1 1						
0 0 1 1	0 1 0 1	0 1 1 1 1						
0 0 1 1	0 1 0 1	0 1 1 1						

References

[1] Eeckman, F. H. and Bower, J. M. (eds.) (1993). Computation and Neural Systems, Kluwer Academic Publishers, Boston.

[2] Bower, J. M. (1992), Modeling the nervous system, *Trends Neurosci.* 15: 411–412.

[3] Bahman Zohuri and Masoud Moghaddam, Business Resilience System (BRS): Driven Through Boolean, Fuzzy Logics and Cloud Computation: Real and Near Real Time Analysis and Decision Making System 1st ed. 2017 Edition.

[4] Robert Hecht-Nielsen, Neurocomputing, Addison-Wesley Publishing Company, 1989.

[5] Rosenblatt, F., "Principle of Neurodynamics," Spartan Book, Washington DE, 1961.

[6] Rosenblatt, F., "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychol. Rev*., **65**, 386-408, 1958.

[7] Steinbuch, K., "Automat und Mensch," Second Edition, Springer-Verlag, Heidelberg, 1963.

[8] Bahman Zohuri, Dimensional Analysis and Self-Similarity Methods for Engineers and Scientists 2015th Edition, Springer Publishing Company.

[9] Stamatios V. Kartalopoulos, "Understanding Neural Networks and Fuzzy Logic, Basic Concepts and Applications," *IEE Press*, 1996.

[10] Rall, W. (1959), Branching dendritic trees and motoneuron membrane resistivity, *Exp. Neurol.* **1**: 491–527.

[11] De Schutter, E. and Bower, J. M. (1994a), An active membrane model of the cerebellar Purkinje cell **I**. Simulation of current clamps in slice, *J. Neurophysiol.* **71**: 375–400.

[12] De Schutter, E. and Bower, J. M. (1994b), An active membrane model of the cerebellar Purkinje cell **II**. Simulation of synaptic responses, *J. Neurophysiol.* **71**: 401–419.

[13] Rapp, M., Yarom, Y. and Segev, I. (1992). The impact of parallel fiber background activity on the cable properties of cerebellar Purkinje cells, *Neural Computation* 4: 518–533.

[14] Segev, I., Rinzel, J. and Shepherd, G. H. (eds) (1995). The Theoretical Foundation of Dendritic Function: Selected Papers by Wilfrid Rall with Commentaries, *MIT Press*, Cambridge, MA.

[15] Principles of Neural Science 4th (fourth) Edition by Kandel, Eric, Schwartz, James, Jessell.

Chapter 7

CABLE AND COMPARTMENTAL MODELS OF DENDRITIC TREES

In modeling neuron, we must deal with two types of complexity, then intricate interplay of active conductances that makes neural dynamics so rich and interesting to study and analyzing them, as well as the elaboration morphology that allows neurons to receive and integrate inputs from so many other neurons. In the previous chapter, we extended the materials, by examining single-compartment models with a wider of voltage-dependent conductances, and hence a wider range of dynamic behaviors, than the Hodgkin-Huxley model. In this chapter, we introduce methods that allow us to study the effects of morphology on the electrical characteristics of neurons. An analytic approach known as cable theory is presented first by Professor David Beeman who wrote this chapter's content and follow up by a discussion of multi-compartment models that permit numerical simulation of complex neuronal structure in a brief form and more details are left in for another book that will be written by these authors in near future. The title of the future book will be the Exploring Realistic Neural Models with the General Neural Simulation System, that Professor Beeman started in the past.

7.1. INTRODUCTION

In the previous chapter, we used a single compartment model to study the mechanisms for the activation of voltage-activated channels, which produce neuron firing. Next, we need to understand how inputs to the neuron affect the potential in the soma and other regions that contain these channels. The following chapter deals with the response of the neuron to synaptic inputs to Produce Postsynaptic Potentials (PSPs). In this chapter, we concentrate on modeling the spread of the PSP through the dendritic tree.

Model neurons range from greatly simplified caricatures to highly detailed descriptions involving thousands of differential equations. Choosing the most appropriate level of modeling for a given research problem requires a careful assessment of the experimental information available and a clear understanding of the research goals. Oversimplified models can of course; give misleading results, but excessively detailed models can obscure interesting results beneath inessential and unconstrained complexity.

An analytic approach known as cable theory is presented first, followed by a discussion of multi-compartment models that permit numerical simulation of complex neuronal structures. Dendrites are strikingly exquisite and unique structures. They are the largest component in both surface area and volume of the brain and their specific morphology is, used to classify neurons into classes: pyramidal, Purkinje, Amacrine cell, stellate, etc. (Figure 7.1). However, most meaningful is that the majority of the synaptic information is, conveyed onto the dendritic tree and it is there where this information is processed. Indeed, dendrites are the elementary computing device of the brain.

A typical dendritic tree receives approximately ten thousand synaptic inputs distributed over the dendritic surface. When activated, each of these inputs produces a local conductance change for specific ions at the postsynaptic membrane, followed by a flow of the corresponding ion current between the two sides of the postsynaptic membrane. As a result, a local change in membrane potential is generated and then spreads along the dendritic branches. How does this spread depend on the morphology (the branching pattern) of the tree and on the electrical properties of its membrane and cytoplasm? This question is a fundamental one; its answer will provide the understanding of how the various synaptic inputs that are, distributed over the dendritic tree interact in time and in space to determine the input-output properties of the neuron and, consequently, their effect upon the computational capability of the neuronal networks that they constitute.

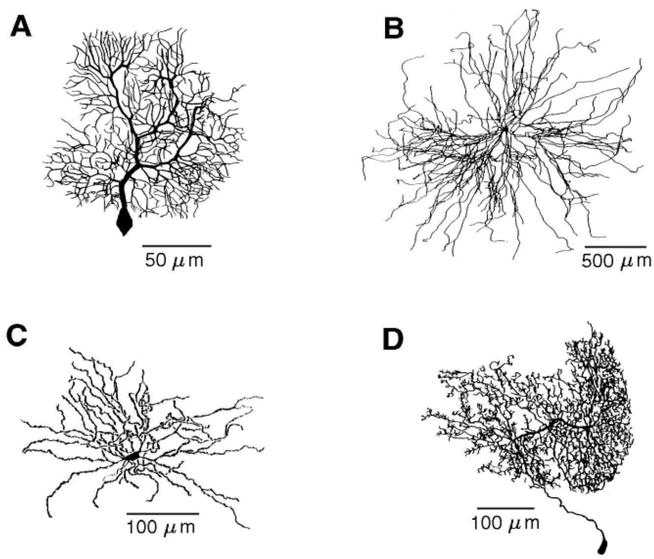


Figure 7.1. Dendrites have unique shapes, which are, used to characterize neurons into types. (A) Cerebellar Purkinje cell of the guinea pig is, reconstructed by Moshe Rapp [1]. (B) a-motoneuron from the cat spinal cord is, reconstructed by Robert Burke [2]. (C) Neostriatal spiny neuron from the rat is, illustrated by Wilson 1992 [3]. (D) Axonless interneurons of the locust is reconstructed by Giles Laurent. In many neuron types, synaptic inputs from a given source are preferentially, mapped into a particular region of the dendritic tree. For example, in Purkinje cells, one excitatory input comes from the vast number of synapses more than 100,000 that specifically contact spines on the thin tertiary branches, whereas the other excitatory input comes from the climbing fiber that contacts the thick (smooth) dendrites. Inhibition from the basket cells impinges close to the Purkinje cell soma, whereas inhibition from stellate cells contacts mainly distal parts of the tree. Note the differences in scales for the different neuron types.

W. Rall developed Cable theory (See the related, section in this chapter) for dendrites in 1959 [4] precisely for this purpose. Namely, to derive a mathematical model that describes the flow of electric current (and the spread of the resultant voltage) in morphologically and physiologically realistic dendritic trees that receive synaptic inputs at various sites and times. In the last thirty years, cable theory for dendrites, complemented by the compartmental modeling approach (Rall 1964) [5], played an essential role in the estimation of dendritic parameters, in designing and interpreting experiments and in providing insights into the computational function of dendrites. This chapter attempts to briefly, summarize cable theory and compartmental models, and to highlight the main results and insights obtained from applying cable and compartmental models to various neuron types. A complete account of Rall's studies, including his principal papers, can be, found in Segev, Rinzel and Shepherd (1995) [6].

We begin by being acquainted with dendrites, the subject matter of this chapter. We then introduce the concepts and assumptions that led to the development of cable theory and then introduce the cable equation. Next, we discuss the implication of this equation for several important theoretical cases as well as for the fully reconstructed dendritic tree. The numerical solution of the cable equation using compartmental techniques is then, presented. We summarize the chapter with the main insights that were, gained from implementing cable and compartmental models with neurons of various types.

7.2. BACKGROUND

In this section, we provide some brief summary of the background before; we proceed with this chapter and topic of cable and compartmental models of dendritic trees.

7.2.1. Dendritic Trees: Anatomy, Physiology and Synaptology

Following the light microscopic studies of the phenomenal neuro-anatomist, Ramón y Cajal, dendrites became the focus of many anatomical investigations and today, with the aid of electron micrograph studies and computer-driven reconstructing techniques, we have a rather intimate knowledge of the fine structure of dendrites. These studies allowed us to obtain essential information on the exact size and type (excitatory or inhibitory) of synaptic inputs as well as on the dimensions of dendrites including the fine structures, the dendritic spines that are involved in the synaptic processing (White 1989 [7], Shepherd 1990 [8]). Here we attempt to introduce, in a concise way, some facts about dendrites and their synaptic inputs. One should remember that, because dendrites come in many shapes and sizes, such a summary unavoidably gives only a rough range of values; more information can be, found in Segev (1995) [9].

1. Branching

Dendrites tend to bifurcate repeatedly and create (often several) large and complicated trees. Cerebellar Purkinje cells, for example, typically bear only one very complicated tree with approximately 400 terminal tips (Figure 7-1A), whereas a-motoneuron from the cat

spinal cord typically possesses 8–12 trees; each has approximately 30 terminal tips (Figure 7-1B). The dendrites of each type of neurons have a unique branching pattern that can be easily, identified and thus help to classify neurons.

2. Diameters

Dendrites are thin tubes of nerve membrane. Near the soma they start with a diameter of a few μm; their diameter typically falls below 1 μm as they successively branch. Many types of dendrites (e.g., cerebellar Purkinje cells, cortical and hippocampal pyramidals) are, studded with abundant tiny appendages, the dendritic spines, which create very thin (~0.1 μm) and short ~ 1 μm) dendritic branches. When present, dendritic spines are the major postsynaptic target for excitatory synaptic inputs and they seem to be important loci for plastic processes in the nervous system (Rall 1964 [5] in Segev et al. 1995 [10], Segev and Rall 1988 [11], Koch and Zador 1993 [12]).

3. Length

Dendritic trees may range from very short (100–200 μm, as in the spiny stellate cell in the mammalian cortex) to quite long (1–2 *mm*, for the spinal a-motoneuron). The total dendritic length may reach 10^4 μm (1 *cm*) and more.

4. Area and Volume

As mentioned in the introduction, the majority of the brain volume and area is, occupied by dendrites. The surface area of a single dendritic tree is in the range of 2,000–750,000 μm^2; its volume may reach up to 30,000 μm^3.

5. Physiology of Dendrites

Both the intracellular cytoplasmic core and the extracellular fluid of dendrites are composed of ionic media that can conduct electric current. The membrane of dendrites can also conduct current via specific transmembrane ion channels, but the resistance to current flow across the membrane is much greater than the resistance along the core. In addition to the membrane channels (membrane resistance), the dendritic membrane can store ionic charges, thus behaving like a capacitor. These $R-C$ properties of the membrane imply a time constant ($\tau_m = RC$) for charging and discharging the transmembrane voltage. The typical range of values for t*m* is 1–100 *msec*. In addition, the membrane and cytoplasm resistivity imply an input resistance (R_{in}) at any given point in the dendritic tree. R_{in} can range from 1 *M*W (at thick and leaky dendrites) and can reach 1000 *M*W (at thin processes, such as dendritic spines). The large values of R_{in} expected in dendrites imply that small excitatory synaptic conductance change (of ~1 *nS*) will produce, locally, a significant (a few tens of *mV*) voltage change. More details of the biophysics of dendrites, the specific properties of their membrane and cytoplasm and their electrical (rather than anatomical) length are, considered below.

In classical cable theory, the assumption was that the electrical properties of the membrane and cytoplasmic properties are *passive* (voltage-independent) so that one could correctly speak of a membrane time *constant* and of a fixed input resistance (at a given site of

the dendritic tree). However, recent recordings from dendrites (e.g., Stuart and Sakmann 1994) [13] clearly demonstrate that the dendritic membrane of many neurons is, endowed with voltage-gated ion channels. This complicates the situation (and makes it more interesting) since, now, the membrane resistivity (and thus τ_m and R_{in}) are voltage-dependent. For sufficiently large voltage perturbations, this nonlinearity may have important consequences on dendritic processing (Rall and Segev 1987). This important issue is further, discussed below.

6. Synaptic Types and Distribution

Synapses are not randomly, distributed over the dendritic surface. In general, inhibitory synapses are more proximal than excitatory synapses, although they are also present at distal dendritic regions and, when present, on some spines in conjunction with an excitatory input. In many systems (e.g., pyramidal hippocampal cells and cerebellar Purkinje cells), a given input source is preferentially mapped onto a given region of the dendritic tree (Shepherd 1990), rather being randomly distributed over the dendritic surface.

The time course of the synaptic conductance change associated with the various types of inputs in a given neuron may vary by 1–2 orders of magnitude. The fast excitatory (AMPA or non-NMDA) and inhibitory (GABAA) inputs operate on a time scale of 1 *msec* and have a peak conductance on the order of 1 *nS*; this conductance is approximately 10 times larger than the slow excitatory (NMDA) and inhibitory (GABAB) inputs that both act on a slower time scale of 10–100 *msec*.

7.2.2. Summary

Dendrites and their spines are the target for a large number of synaptic inputs that, in many cases, are non-randomly distributed over the dendritic surface. The dendritic membrane is equipped with a variety of synaptically activated and voltage-gated ion channels. The kinetics and voltage dependence of these channels, together with a particular dendritic morphology and input distribution, make the dendritic tree behave as a complex *dynamical* device with a potentially rich repertoire of computational (input-output) capabilities. Cable theory for dendrites provides the mathematical framework that enables one to connect the morphological and electrical structure of the neuron to its function.

7.3. The One-Dimensional Cable Equation

In this section, we will discuss, the One-Dimensional Cable Equation by driving it and then solve simple second-order partial differential equation for its solution.

7.3.1. The One-Dimensional Cable Equation

As mentioned previously, dendrites are thin tubes wrapped with a membrane that is a relatively good electrical insulator compared to the resistance provided by the intracellular

core or the extracellular fluid. Because of this difference in membrane versus axial resistivity, for a short length of dendrite, the electrical current inside the core conductor tends to flow parallel to the cylinder axis (along the x-axis). This is why the classical cable theory considers only one spatial dimension (x) along the cable, while neglecting the y and z dimensions. In other words, one key assumption of the one-dimensional cable theory is that the voltage V across the membrane is a function of only time t and distance x along the core conductor.

The other fundamental assumptions in the classical cable theory are:

1. The membrane is passive (voltage-independent) and uniform.
2. The core conductor has constant cross section and the intracellular fluid can be, represented as an ohmic resistance.
3. The extracellular resistivity is negligible (implying extracellular isopotentiality).
4. The inputs are currents (which sum linearly, in contrast to changes in synaptic membrane conductance, whose effects do not sum linearly, as it can be, seen in Chapter 6 of the book by Bower and Beeman).

For convenience, we will make the additional assumption that the membrane potential is, measured with respect to a resting potential of zero, as we assumed in the previous chapter.

These assumptions allow us to write down the one-dimensional passive cable equation for $V(x,t)$, the voltage across the membrane cylinder at any point x and time t. As was shown by Rall (1959) [4], this equation can be, solved analytically for arbitrarily complicated passive dendritic trees. As noted before, real dendritic trees receive conductance inputs (not current inputs) and may possess nonlinear membrane channels (violating the passive assumption). Yet, as we discuss later, the passive case is very important as the essential reference case, and it provided the fundamental insights regarding signal processing in dendrites.

7.3.2. The Cable Equation

At any point along a cylindrical membrane segment (core conductor), current can flow either longitudinally (along the x-axis), or through the membrane. The longitudinal current I_i (in amperes) encounters the cytoplasm resistance, producing a voltage drop. We take this current to be positive when flowing in the direction of increasing values of x, and define the *cytoplasm resistivity* as a resistance per unit length along the *x*-axis *ri*, expressed in units of Ω/*cm*. Then, Ohm's law allows us to write

$$\frac{1}{r_i}\frac{\partial V}{\partial x} = -I_i \qquad \text{Eq. 7-1}$$

The membrane current can either cross the membrane via the passive (resting) membrane channels, represented as a resistance r_m (in Ω *cm*) for a unit length, or charge (discharge) the membrane capacitance per unit length c_m (in *F/cm*). If no additional current is applied from an electrode, then the change per unit length ($\partial I_i / \partial x$) of the longitudinal current must be the density of the membrane current i_m per unit length (taken positive outward),

$$\frac{\partial I_i(x,t)}{\partial x} = -i_m = -\left(\frac{V(x,t)}{r_m} + c_m \frac{\partial V(x,t)}{\partial t}\right) \quad \text{Eq. 7-2}$$

Combining Equation 7-1 and Equation 7-2, we get the cable equation, a second-order Partial Differential Equation (PDE) for $V(x,t)$

$$\frac{1}{r_i}\frac{\partial^2 V(x,t)}{\partial x^2} = c_m \frac{\partial V(x,t)}{\partial t} + \frac{V(x,t)}{r_m} \quad \text{Eq. 7-3}$$

For the derivation of Equation 7-3, it has been useful to consider the cytoplasm resistivity r_i, membrane resistivity r_m and membrane capacitance *cm* for a unit length of cable having some fixed diameter. If we want to describe the cable properties in terms of the cable diameter, or we wish to make a compartmental model of a dendrite based on short sections of length *l* (Section 7.5), we will need expressions for the actual resistances and capacitance in terms of the cable dimensions.

It is often useful to refer to the membrane capacitance or resistance of a patch of membrane that has an area of 1 cm^2, so that our calculations can be independent of the size of a neural compartment. These quantities are, called the *specific capacitance* and *specific resistance* of the membrane. In the book, and in the GENESIS [15] tutorials, we denote the specific capacitance by C_M and the specific resistance by R_M, and use the symbols C_m and R_m for the actual values of the membrane capacitance and resistance of a section of dendritic cable in farads and ohms. This can be a point of confusion when reading other descriptions of cable theory, as it is also common to use the same notation (C_m and R_m) for the specific quantities.

The capacitance of biological membranes was found to have a specific value C_M close to 1 $\mu F/cm^2$. Hence, the actual capacitance C_m of a patch of cylindrical membrane with diameter d and length l is $\pi d / C_M$. In terms of the capacitance per unit length, $C_m = 1\ c_m$. If the passive channels are uniformly distributed over a small patch of membrane, the conductance will be proportional to the membrane area. This means that the membrane resistance will be inversely proportional to the area and that it can be written as

$R_m = R_M/(\pi l d)$, or as $R_m = r_m/l \cdot R_M$ is then expressed in units of Ωcm². Later in this chapter, we perform some simulations in which a number of cylindrical compartments are connected through their axial resistances R_a.

As this resistance is proportional to the length of the compartment and inversely proportional to its cross sectional area, we can define a *specific axial resistance* R_A that is independent of the dimensions of the compartment and has units of Ω *cm*. Thus, a cylindrical segment of length l and diameter d will have an axial resistance R_a of $(4/R_A)/\pi d^2$, or lr_i.

We can summarize these relationships with the equations

$$C_m = c_m l = \pi l d C_M \qquad \text{Eq. 7-4}$$

$$R_m = \frac{r_m}{l} = \frac{R_M}{\pi l d} \qquad \text{Eq. 7-5}$$

and

$$R_a = r_i l = \frac{4 l R_A}{\pi d^2} \qquad \text{Eq. 7-6}$$

It is useful to define the *space constant*,

$$\lambda = \sqrt{r_m/r_i} = \sqrt{(d/4) R_M / R_A} \qquad \text{Eq. 7-7}$$

(in *cm*) and the *membrane time constant*,

$$\tau_m = r_m c_m = R_M C_M = R_m C_m \qquad \text{Eq. 7-8}$$

Then, the cable Equation 7-3 becomes

$$\lambda^2 \frac{\partial^2 V}{\partial x^2} - \tau_m \frac{\partial V}{\partial t} - V = 0 \qquad \text{Eq. 7-9}$$

or in dimensionless units,

$$\frac{\partial^2 V}{\partial X^2} - \frac{\partial V}{\partial T} - V = 0 \qquad \text{Eq. 7-10}$$

where $X = x/\lambda$ and $T = t/\tau_m$ [16]. A complete derivation of the cable equation can be, found in Rall (1989) [17] and in Jack, Noble and Tsien (1975) [18]. However, in next section we show brief steps toward derivation of the Cable Equation here.

7.3.3. Derivation of the Cable Equation

We are going to use the following image, as Figure 7-2 in order to derive the Cable Equation here.

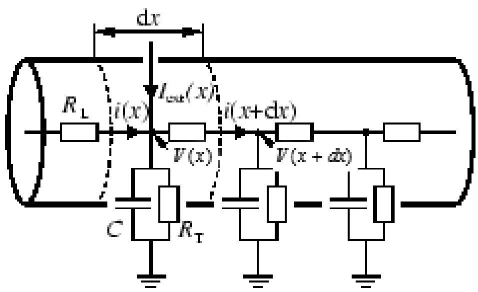


Figure 7-2. Part of a Dendrite and the Corresponding Circuit Diagram.

In Figure 7-2, longitudinal and transversal resistors are denoted by R_L and R_T, respectively. The electrical capacity of each small piece of dendrite is, symbolized by capacitor C. Note that, dendrite, is short, branched extension of a nerve cell, along which impulses received from other cells at synapses are, transmitted to the cell body.

Consider a piece of a dendrite decomposed in short cylindric segments of length dx each. The schematic drawing in Figure 7-2 shows the corresponding circuit diagram. Using Kirchhoff's law, we find equations that relate the voltage $V(x)$ across the membrane at location x with longitudinal and transversal currents. First, a longitudinal current $i(x)$ passing through the dendrite causes a voltage drop across the longitudinal resistor R_L according to Ohm's law

$$V(x+dx,t) - V(x) = R_L i(x,t) \qquad \text{Eq. 7-11}$$

where $V(x+dx,t)$ is the potential at the neighboring point $x+dx$. Second, the transversal current that passes through the RC-circuit is given by $C\partial V(x,t)/\partial t + V(x,t)/R_T$. Kirchhoff's law regarding the conservation of current at each node leads to

$$i(x+dx,t) - i(x,t) = C\frac{\partial V(x,t)}{\partial t} + \frac{V(x,t)}{R_T} - I_{ext}(x,t) \qquad \text{Eq. 7-12}$$

The values of the longitudinal resistance R_L, the transversal conductivity $1/R_T$, the capacity C, and the externally applied current can be, expressed in terms of specific quantities per unit length r_L/r_T, c, and $i_{ext}(x,t)$, respectively, via:

$$R_L = r_L dx\,,\ \frac{1}{R_T} = \frac{1}{r_T}dx\,,\quad C = cdx\,,\qquad I_{ext}(x,t) = i(x,t)dx$$

Eq. 7-13

These scaling relations express the fact that the longitudinal resistance and the capacity increase with the length of the cylinder, whereas the transversal resistance is decreasing, simply because the surface the current can pass through is increasing. Substituting, these expressions in Equations 7-12 and 7-13, dividing by dx, and taking the limit $dx \to 0$ leads to

$$\frac{\partial V(x,t)}{\partial x} = r_L i(x,t)$$ Eq. 7-14(a)

$$\frac{\partial i(x,t)}{\partial x} = c\frac{\partial V(x,t)}{\partial t} + \frac{V(x,t)}{r_T} - i_{ext}(x,t)$$ Eq. 7-14(b)

Taking the derivative of these equations with respect to χ and crosswise substitution yields

$$\frac{\partial^2 V(x,t)}{\partial x^2} = cr_L\frac{\partial V(x,t)}{\partial t} + \frac{r_L}{r_T}V(x,t) - r_L i_{ext}(x,t)$$ Eq.7-15(a)

$$\frac{\partial^2 i(x,t)}{\partial x^2} = cr_L\frac{\partial i(x,t)}{\partial t} + \frac{r_L}{r_T}i(x,t) - \frac{\partial i_{ext}(x,t)}{\partial x}$$ Eq. 7-15(b)

We are going to take advantages of dimensional analysis [16], by introducing the characteristic length scale $\lambda^2 = r_T/r_L$ as electrotonic length scale, and the membrane time constant $\tau = r_T c$. If we multiply both form of Equations 7-15 by electrotonic length scale λ^2, we obtain

$$\lambda^2\frac{\partial^2 V(x,t)}{\partial x^2} = \tau\frac{\partial V(x,t)}{\partial t} + V(x,t) - r_T i_{ext}(x,t)$$ Eq. 7-16(a)

$$\lambda^2 \frac{\partial^2 i(x,t)}{\partial x^2} = \tau \frac{\partial i(x,t)}{\partial t} + i(x,t) - \frac{r_T}{r_L} \frac{\partial i_{ext}(x,t)}{\partial x} \quad \text{Eq. 7-16(b)}$$

After a transformation to unit-free coordinates

$$x \to \hat{x} = x/\lambda$$

and

$$i_{ext} \to \hat{i}_{ext} = r_T i_{ext} \quad \text{Eq. 7-17}$$

We some mathematical manipulation, we obtain the cable equations as follows;

$$\frac{\partial V(x,t)}{\partial t} = \frac{\partial^2 V(x,t)}{\partial x^2} - V(x,t) + i_{ext}(x,t) \quad \text{Eq. 7-18(a)}$$

$$\frac{\partial i(x,t)}{\partial t} = \frac{\partial^2 i(x,t)}{\partial x^2} - i(x,t) + \frac{\partial i_{ext}(x,t)}{\partial x} \quad \text{Eq. 7-19(b)}$$

in a symmetric, unit-free form. Note that it suffices to solve one of these equations due to the simple relation between *u* and *i* given in Equation 7-15(a).

The cable equations can be easily, interpreted. These equations describe the change in time of voltage and longitudinal current. Both equations contain three different contributions. The first term on the right-hand side of Equation 7-25 further down in the chpater, is a diffusion term that is positive if the voltage (or current) is a convex function of x. The voltage at *x* thus tends to decrease, if the values of V are lower in a neighborhood of x than at x itself. The second term on the right-hand side of Equation 7-25 is a simple decay term that causes the voltage to decay exponentially towards zero. The third term, finally, is a source term that acts as an inhomogeneity in the otherwise autonomous differential equation. This source can be due to an externally applied current, to synaptic input, or to other (non-linear) ion channels; Section 7.4.6.

7.4. Solution of the Cable Equation for Several Cases

In the following sub-sections, we attempt to solve the Partial Differential Equation PDE of cable equation with various boundary conditions and other related possibilities of the form of PDE.

7.4.1. Steady-State Voltage Attenuation with Distance

The solution of Equation 7-10 depends, in addition to the electrical properties of the membrane which and cytoplasm, on the initial condition and the boundary condition at the end of the segment toward the current flows. Consider the simple case of a steady state ($\partial V/\partial T = 0$); the cable Equation 7-10 is, reduced to an ordinary differential equation.

$$\frac{\partial^2 V}{\partial X^2} - V = 0 \qquad \text{Eq. 7-20}$$

Whose general solution can be expressed as;

$$V(X)\big|_{X=x/\lambda} = Ae^{X} + Be^{-X} \qquad \text{Eq. 7-21}$$

where *A* and *B* depend on the boundary conditions. In the case of a cylindrical segment of infinite extension, where $V(X) = 0$ at $X = \infty$, and $V(X) = V_0$ at $X = 0$, the solution for Equation 7-11 is

$$V(X)\big|_{X=x/\lambda} = V_0 e^{-X} = V_0 e^{-x/\lambda} \qquad \text{Eq. 7-22}$$

Thus, in this case, the steady voltage attenuates exponentially with distance. Indeed, in a very long uniform cylindrical segment, a steady voltage attenuates e-fold for each unit of λ. This holds only for a cylinder of infinite length.

Now, let us consider a finite length of dendritic cable. If it has a length 1, we can define the dimensionless *electrotonic length* as $L = 1/\lambda$. When the cylindrical segment has a sealed end at $X = L$ ("open circuit termination"), no longitudinal current flows at this end. Then, the solution for Equation 7-20 with $V = V_0$ at $X = 0$ is

$$V = \frac{V_0 \cosh(L - X)}{\cosh(L)}, \qquad \text{for} \qquad \frac{\partial V(X)}{\partial X} = 0 \text{ at } X = L$$

Eq. 7-23

As shown in Figure 7-3 here, in finite cylinders with sealed ends, steady voltage attenuates less steeply than $\exp(-x/\lambda)$. In the other extreme, where the point $X = L$ is clamped to the resting potential, which denote here, for simplicity, as 0, the solution to Equation 7-20 becomes

$$V = \frac{V_0 \sinh(L - X)}{\sinh(L)}, \qquad \text{for} \qquad V(X) = 0 \text{ at } X = L$$

Eq. 7-24

In this case, steady voltage attenuates more steeply than the attenuation in the infinite case (Figure 7-3). In dendritic trees, a dendritic segment typically ends with a sub-tree; this "leaky end" condition is somewhere between the sealed end (Equation 7-23) and the "clamped to rest" condition (Equation 7-24). See Section 7.4.10.

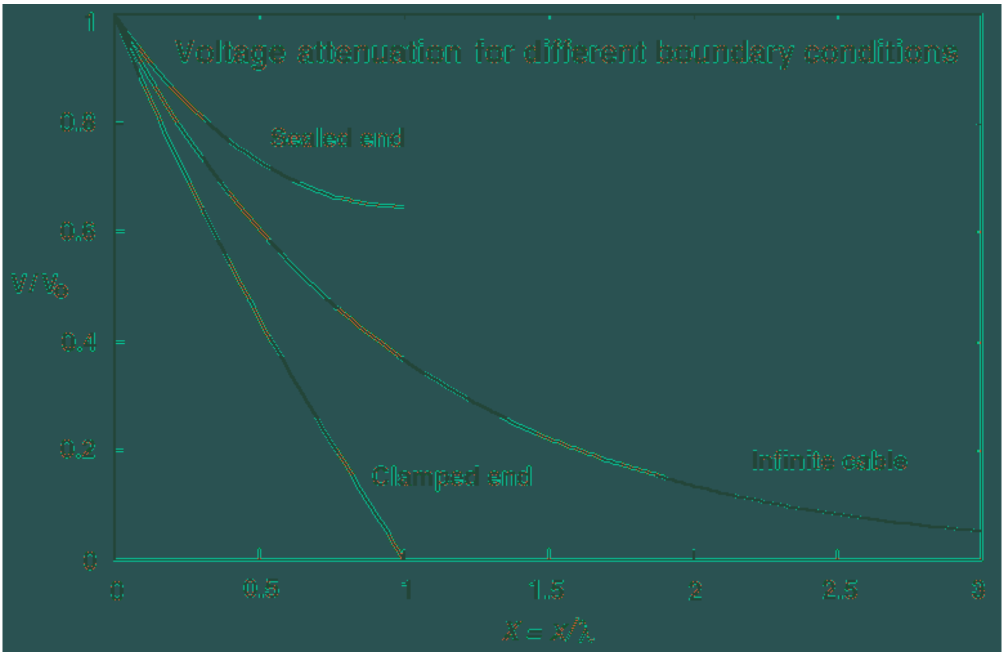


Figure 7-3. Attenuation of Membrane Potential with Distance in a Cylindrical Cable with Different Boundary Conditions.

The middle curve shows the attenuation in an infinite cable (Equation 7-23). The other two plots are for a finite length cable of electrotonic length $L=1$. The upper one is for a sealed end cable (no current flows past the end, Equation 7-23) and the lower is for a cable with the end clamped to the resting potential of 0 (Equation 7-24).

7.4.2. Stationary Solutions of the Cable Equation

In order to get an intuitive understanding of the behavior of the cable equation we look for stationary solutions of type as Equation 7-25 in below, i.e., for solutions with $\{\partial V(x,t)/\partial t\}=0$. In that case, the partial differential equation 7-18(a), reduces to an ordinary differential equation in χ as:

$$\frac{\partial^2 V(x,t)}{\partial x^2} - V(x,t) = -i_{ext}(x,t) \quad \text{Eq. 7-25}$$

The general solution to the homogeneous equation with $i_{ext}(x,t) \equiv 0$, is as follow:

$$V(x,t) = A\sinh(x) + B\cosh(x) \qquad \text{Eq. 7-26}$$

As can easily be checked by taking the second derivative with respect to x. Here, A and B are constants that are determined by the boundary conditions. Solutions for non-vanishing input current can be, found by standard techniques. For a stationary input current $i_{ext}(x,t) = \delta(x)$ localized at $x = 0$ and boundary conditions $V(\pm\infty) = 0$, we find (See Figure 7-4)

$$V(x,t) = \frac{1}{2}e^{-|x|} \qquad \text{Eq. 7-27}$$

This solution is given in units of the intrinsic length scale $\lambda = (r_T/r_L)^{1/2}$. If we re-substitute the physical units, we see that λ is the length over, which the stationary membrane potential drops by a factor $1/e$. In the literature λ is, referred to as the electrotonic length scale (Rall, 1989) [19]. Typical values for the specific resistance of intracellular medium and the cell membrane are 100 Ωcm and 30 kΩcm^2, respectively. In a dendrite with radius $\rho = 1$ µm this amounts to a transversal and a longitudinal resistance of $r_L = 100\,\Omega\text{cm}/(\pi\rho^2) = 3\cdot 10^5\,\Omega\mu m^{-1}$ and $r_L = 30\,\text{k}\Omega\text{cm}^2/(2\pi\rho) = 5\cdot 10^{11}\,\Omega\mu m$. The corresponding electrotonic length scale is λ = 1.2 mm. Note that the electrotonic length can be significantly smaller if the transversal conductivity is increased, e.g., due to open ion channels.

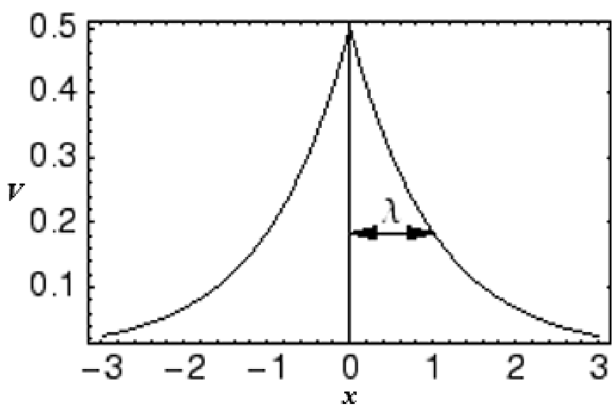


Figure 7-4. Stationary Solution of the Cable Equation with a Constant Current of Unit Strength.

Note that, in Figure 7-4, the constant current of unit strength being injected at $x = 0$, i.e., $i_{ext}(x,t) = \delta(x)$. The electrotonic length scale λ is the distance over which the membrane potential drops to $1/e$ of its initial value.

For arbitrary stationary input current $i_{ext}(x)$ the solution of Equation 7-25 can be, obtained by a superposition of translated fundamental solutions as Equation 7-28, that is shown here:

$$V(x,t) = \frac{1}{2}\int e^{-|x-x'|} i_{ext}(x')dx' \qquad \text{Eq. 7-28}$$

This is an example of the Green's function approach applied here to the stationary case. The general time-dependent case will be, treated in the next section.

7.4.3. Green's Function

In the following, we will concentrate on the equation for the voltage and start our analysis by deriving the Green's function for a cable extending to infinity in both directions. The Green's function is, defined as the solution of a linear equation such as Equation 7-25, with a Dirac δ -pulse as its input. It can be, seen as an elementary solution of the differential equation because - due to linearity - the solution for any given input can be, constructed as a superposition of these Green's functions.

In order to find the Green's function for the cable equation we thus have to solve Equation 7-25 with $i_{ext}(x,t)$ replaced by a δ impulse at $x = 0$ and $t = 0$,

$$\frac{\partial V(x,t)}{\partial t} - \frac{\partial^2 V(x,t)}{\partial x^2} + V(x,t) = \delta(x)\delta(t) \qquad \text{Eq. 7-29}$$

Fourier transformation with respect to the spatial variable yields

$$\frac{\partial V(k,t)}{\partial t} - \frac{\partial^2 V(k,t)}{\partial x^2} + V(k,t) = \frac{\delta(t)}{\sqrt{2\pi}} \qquad \text{Eq. 7-30}$$

This is an ordinary differential equation in t and has a solution of the form

$$V(k,t) = \exp\left\{\frac{-(1+k^{2)t}}{\sqrt{2\pi}}\right\}\Theta(t) \qquad \text{Eq. 7-31}$$

With $\Theta(t)$ denoting the Heaviside function. After an inverse Fourier we obtain the desired Green's function $G_\infty(x,t)$, and

$$V(x,t) = \frac{\Theta(t)}{\sqrt{4\pi t}} \exp\left[-t - \frac{x^2}{4t}\right] \equiv G_\infty(x,t) \qquad \text{Eq. 7-32}$$

The general solution for an infinitely long cable is therewith, given through

$$V(x,t) = \int_{-\infty}^{t} dt' \int_{-\infty}^{+\infty} dx' G_\infty(x-x', t-t') i_{ext}(x',t') \qquad \text{Eq. 7-33}$$

7.4.4. Checking the Green's Property

We can check the validity of Equation 7-33 by substituting $G_\infty(x,t)$ into the left-hand side of Equation 7-30. After a short calculation and manipulation, we find that:

$$\left[\frac{\partial}{\partial t} - \frac{\partial^2}{\partial x^2} + 1\right] G_\infty(x,t) = \frac{1}{\sqrt{4\pi t}} \exp\left(-t - \frac{x^2}{4t}\right) \delta(t) \qquad \text{Eq. 7-34}$$

Where we have used $\partial\Theta/\partial t = \delta(t)$. As long as $t \neq 0$ the right-hand side of Equation 7-35 below vanishes, as required by Equation 7-33. For $t \to 0$, we find that

$$\lim_{t\to 0} \frac{1}{\sqrt{4\pi t}} \exp\left(-t - \frac{x^2}{4t}\right) = \delta(x) \qquad \text{Eq. 7-35}$$

which proves that the right-hand side of Equation 7-35 is indeed equivalent to the right-hand side of Equation 7-33. Having established that, we can then write the following relation as:

$$\left[\frac{\partial}{\partial t} - \frac{\partial^2}{\partial x^2} + 1\right] G_\infty(x,t) = \delta(x)\delta(t) \qquad \text{Eq. 7-36}$$

We can readily show that Equation 7-34 is the general solution of the cable equation for an arbitrary input currents $i_{ext}(x_0, t_0)$. We now, substitute Equation 7-34 into the cable equation, exchange the order of integration and differentiation, and find that:

$$\begin{aligned}
&\left[\frac{\partial}{\partial t} - \frac{\partial^2}{\partial x^2} + 1\right] V(x,t) \\
&\quad = \int_{-\infty}^{t} dt' \int_{-\infty}^{+\infty} dx' \left[\frac{\partial}{\partial t} - \frac{\partial^2}{\partial x^2} + 1\right] G_\infty(x-x', t-t') \\
&\quad = \int_{-\infty}^{t} dt' \int_{-\infty}^{+\infty} dx' \delta(x-x')\delta(t-t') G_\infty(x-x', t-t') i_{ext}(x',t') \\
&\quad = i_{ext}(x,t)
\end{aligned} \qquad \text{Eq. 7-37}$$

and solution is verified.

7.4.5. Finite Cable

Real cables do not extend from $-\infty$ to $+\infty$ and we have to take extra care to, correctly include boundary conditions at the ends. We consider a finite cable extending from $x=0$ to $x=L$ with sealed ends, i.e., $i(x=0,t)=i(x=L,t)=0$ or, equivalently, $\frac{\partial}{\partial x}V(x=0,t)=\frac{\partial}{\partial x}V(x=L,t)=0$.

The Green's function $G_{0,L}$ for a cable with sealed ends can be constructed from G_∞ by applying a trick from electro-statics called “mirror charges” (Jackson, 1962) [20]. Similar techniques can also be applied to treat branching points in a dendritic tree (Abbott, 1991) [21]. The cable equation is linear and, therefore, super-positions of two solutions are solution as well. Consider a δ current pulse at time t_0 and position x_0 somewhere along the cable. The boundary condition $\frac{\partial}{\partial x}V(x=0,t)=0$ can be satisfied if we add a second, virtual current pulse at a position $x=-x_0$ *outside* the interval $[0,L]$. Adding a current pulse outside the interval $[0,L]$ comes for, free since the result is still a solution of the cable equation on that interval. Similarly, we can fulfill the boundary condition at $x = L$ by adding a mirror pulse at $x=2L-x_0$. In order to account for both boundary conditions simultaneously, we have to compensate for the mirror pulse at $-x_0$ by adding another mirror pulse at $2L + x_0$ and for the mirror pulse at $x=2L-x_0$ by adding a fourth pulse at $-2L+x_0$ and so forth. Altogether, we have

$$G_{0,L}(x_0,t_0:x,t)=$$

$$n=\sum_{n=-\infty}^{\infty}\{G_\infty(x-2nL-x_0,t-t_0)+G_\infty(x-2nL+x_0,t-t_0)\}$$

Eq. 7-38

We emphasize that in the above Green's function we have to specify both (t_0, x_0) and (t, x) because the setup is no longer translation invariant. The general solution on the interval $[0,L]$ is, given by

$$V(x,t)=\int_{-\infty}^{t}dt_0\int_0^L dx_0 G_0(t,x:t_0,x_0)i_{ext}(x_0,t_0) \qquad \text{Eq. 7-39}$$

An example for the spatial distribution of the membrane potential along the cable is shown in Figure 7-5 A, where a current pulse has been injected at location $x=1$. In addition to Figure 7-5A, subfigure B exhibits the *time course* of the membrane potential measured in various distances from the point of injection. It is clearly visible that the peak of the membrane potential measured at, e.g., $x=3$ is more delayed than at, e.g., $x=2$. In

addition, the amplitude of the membrane potential decreases significantly with the distance from the injection point.

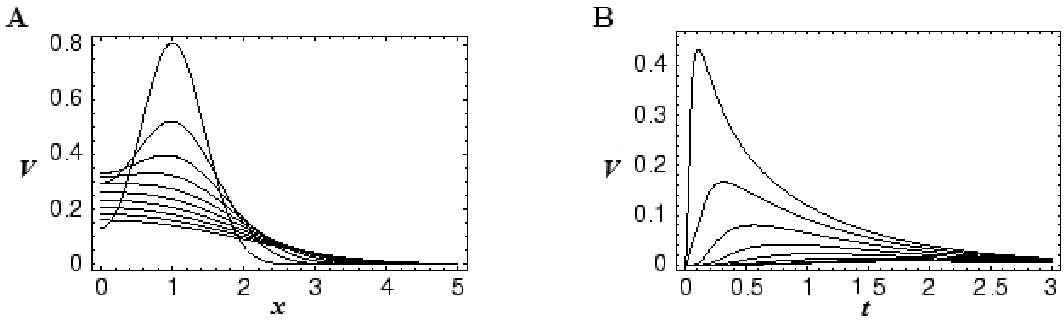


Figure 7-5. Exhibition of Time Course of the Membrane Potential Measured in Various Distances from the Point of Injection.

This well-known phenomenon is also present in neurons. In the absence of active amplification mechanisms, synaptic input at distal dendrites produces broader and weaker response at the soma as compared to synaptic input at proximal dendrites.

7.4.6. Non-Linear Extensions of the Cable Equation

In the context of a realistic modeling of `biological' neurons, two non-linear extensions of the cable equation have to be, discussed. The obvious one is the inclusion of non-linear elements in the circuit diagram of Figure 7-2 that account for specialized ion channels. As we have seen in the Hodgkin-Huxley model, ion channels can exhibit a complex dynamics that is in itself governed by a system of (ordinary) differential equations. The current through one of these channels is thus not, simply a (non-linear) function of the actual value of the membrane potential but may also depend on the time course of the membrane potential in the past. Using the symbolic notation $i_{ion}[V](x,t)$ for this functional dependence the extended cable equation takes the form

$$\frac{\partial}{\partial t}V(x,t)=\frac{\partial^2}{\partial x^2}V(x,t)-V(x,t)-i_{ion}[V](x,t)+i_{ext}(x,t) \qquad \text{Eq. 7-40}$$

A more subtle complication arises from the fact that a synapse cannot be, treated as an ideal current source. The effect of an incoming action potential is the opening of ion channels. The resulting current is proportional to the difference of the membrane potential and the corresponding ionic reversal potential. Hence, a time-dependent conductivity as in Equation 7-41 below provides a more realistic description of synaptic input than an ideal current source with a fixed time course.

Equation 7-41, is, derived, based on the current that passes through two classes of ion channel, namely *voltage*-activated and *calcium*-activated ion channels and, thus, in an excitatory or inhibitory postsynaptic current (EPSC or IPSC) as written here:

$$I_{syn}(t) = g_{syn}(t)(V - E_{syn}) \qquad \text{Eq. 7-41}$$

The parameter E_{syn} and the function $g_{syn}(t)$ can be, used to characterize different types of synapse. Typically, a superposition of exponentials is used for $g_{syn}(t)$. For inhibitory synapses E_{syn} equals the reversal potential of potassium ions (about -75 mV), whereas for excitatory synapses $E_{syn} \approx 0$.

Note that: So far, we have encountered two classes of ion channel, namely *voltage*-activated and *calcium*-activated ion channels. A third type of ion channel we have to deal with is that of *transmitter*-activated ion channels involved in synaptic transmission. Activation of a presynaptic neuron results in a release of neurotransmitters into the synaptic cleft. The transmitter molecules diffuse to the other side of the cleft and activate receptors that are located in the postsynaptic membrane. So-called *ionotropic receptors* have a direct influence on the state of an associated ion channel whereas *metabotropic receptors* control the state of the ion channel by means of a biochemical cascade of g-proteins and second messengers. In any case the activation of the receptor results in the opening of certain ion channels and, thus, in an excitatory or inhibitory postsynaptic current (EPSC or IPSC). This falls under the understanding that we have had so far, about detailed Neuron Models. From a biophysical point of view, action potentials are the result of currents that pass through ion channels in the cell membrane. In an extensive series of experiments on the giant axon of the squid, Hodgkin and Huxley succeeded to measure these currents and to describe their dynamics in terms of differential equations.

The Hodgkin-Huxley equations are the starting point for detailed neuron models, which account for numerous ion channels, different types of synapse, and the specific spatial geometry of an individual neuron. Ion channels, synaptic dynamics, and the spatial structure of dendrites are the topics of this chapter and pervious one as well.

If we replace in Equation 7-25 the external input current $i_{ext}(x,t)$ by an appropriate synaptic input current $-i_{syn}(x,t) = -g_{syn}(x,t)[V(x,t) - E_{syn}]$ with g_{syn} being the synaptic conductivity and E_{syn} the corresponding reversal potential, we obtain:

$$\frac{\partial}{\partial t}V(x,t) = \frac{\partial^2}{\partial x^2}V(x,t) - V(x,t) - g_{syn}(x,t)[V(x,t) - E_{syn}] \qquad \text{Eq. 7-42}$$

This is still a linear differential equation but its coefficients are now time-dependent. If the time course of the synaptic conductivity can be, written as a solution of a differential equation then the cable equation can be, reformulated so that synaptic input reappears as an

inhomogeneity to an autonomous equation. For example, if the synaptic conductivity is simply, given by an exponential decay with time constant τ_{syn} we have the following relationship as:

$$\frac{\partial}{\partial t}V(x,t)-\frac{\partial^2}{\partial x^2}V(x,t)+V(x,t)+g_{yn}(x,t)[V(x,t)-E_{syn}]=0 \qquad \text{Eq. 7-43a}$$

$$\frac{\partial}{\partial t}g_{yn}(x,t)-\frac{g_{syn}(x,t)}{\tau_{syn}}=S(x,t) \qquad \text{Eq. 7-43b}$$

Here, $S(x,t)$ is a sum of Dirac δ functions which describe the presynaptic spike train that arrives at a synapse located at position *x*. Note that this equation is *non*-linear because it contains a product of g_{syn} and V which are both unknown functions of the differential equation. Consequently, the formalism based on Green's functions cannot be, applied.

7.4.7. Voltage Decay with Time

Consider the other extreme case of the cable equation (7-10) where $\partial V/\partial x=0$. The cable is “shrunken” to an isopotential element, and Equation 7-10 is, reduced to an Ordinary Differential Equation (ODE),

$$\frac{dV(T)}{dT}+V(T)=0 \qquad \text{Eq. 7-44}$$

Whose general solution can be expressed as

$$V(T)=Ae^{-T} \qquad \text{Eq. 7-45}$$

where A depends on the initial condition. When a current step I_{pulse} is, injected to this isopotential neuron, the resultant voltage $V(T)$ is

$$V(T)=I_{pulse}R_m(1-e^{-T})=I_{pulse}R_m(1-e^{-t/\tau_m}) \qquad \text{Eq. 7-46}$$

where R_m is the net membrane resistance (in ohms) of this isopotential segment. At the cessation of the current at $t=t_0$, the voltage decays exponentially from its maximal value $V_0=V(t_0)$;

$$V(T) = V_0 e^{-T} = V_0 e^{-t/\tau_m}, \quad \text{for} \quad t \geq t_0$$

Eq. 7-47

Equation 7-46 implies that, because of the R-C properties of the membrane, the voltage developed as a, result of the current input lags behind the current input; Equation 7-47 implies that voltage remains for some time after the input ends ("memory"). For more details, see Bower and Beeman [15].

In the general case of passive cylinders, the solution to the cable equation (Equation 7-10) can be, expressed as a sum of an infinite number of exponential decays,

$$V(X,T) = \sum_{t=0}^{\infty} B_i \cos\left(i\pi X/L\right) e^{-t/\tau_i}$$

Eq. 7-48

where the Fourier coefficients B_i depend only on L, the index i, and on the initial conditions (the input point and the initial distribution of voltage over the tree). The time constants τ_i are independent of location in the tree; $\tau_i < \tau_{i+1}$ for any i and, for the uniform membrane, the slowest time constant $\tau_0 < \tau_m$. The shorter ("equalizing") time constants (τ_i , for $i = 1, 2, \cdots$) depend only on the electrotonic length *L* of the cylinder (in units of l). Specifically, in cylinders of length *L* with a sealed end, they are, given by

$$\frac{\tau_0}{\tau_i} = 1 + \left(\frac{i\pi}{L}\right)^2$$

Eq. 7-49

Consequently, Rall showed that *L* can be estimated directly from the values of t*i*, in particular from the two slowest time constants, t0 and t1, that can be "peeled" from the experimental voltage transient (Rall 1969) [22]. More details are, given in reviews by Rall (1977 [23], 1989 [19]) and Jack et al. (1975).

Rall also showed that the time course of synaptic potentials changes as the recording point moves away from the input location. The time course of the voltage response near the input site is relatively rapid and it becomes significantly prolonged (and attenuated) at a point distant from the input site. This effect is the source of Rall's method (Rall 1967) [24] of using shape parameters of the synaptic potentials (its "rise time" and its width at half amplitude, the "half-width") to estimate the electrotonic distance of the synapse (the input) from the soma (the recording site).

7.4.8. Functional Significance of λ and τ_m

The space constant l and the membrane time constant t*m* are two very important parameters that play a critical role in determining the spatio-temporal integration of synaptic

inputs in dendritic trees. Equation 7-47 shows that τ_m provides a measure of the time window for input integration. A cell with large τ_m (e.g., 50 *msec*) integrates inputs over a larger time window compared to cells with smaller t*m* values (say, 5 *msec*). The value of t*m* depends on the electrical properties of the membrane R_M and C_M , but it does not depend on the cell morphology. Neurons with a high density of open membrane channels (i.e., with a small *RM* value) will respond quickly to the input and will "forget" this input rapidly. In contrast, neurons with relatively few open membrane channels (large R_M will be able to summate inputs for relatively long periods of time (slow voltage decay).

In contrast to τ_m, the space constant λ depends not only on the membrane properties but also on the specific axial resistance and the diameter. In neurons with large l (e.g., with large R_M and/or large diameter, or small R_A) voltage attenuates less with distance as compared to neurons with a smaller l value. Thus, in the former case, inputs that are anatomically distant from each other will summate better (spatially) with each other than in the latter case. Therefore, knowledge of λ and τ_m for a, given neuron provides important information about the capability of its dendritic tree to integrate inputs both in time and in space.

7.4.9. The Input Resistance R_{in} and "Trees Equivalent to a Cylinder"

A third important parameter is R_{in}, the *input resistance* at a given point in the dendritic tree. When a steady current I_0 is applied at a given location in a structure, a steady voltage V_0 is developed at that point. The ratio V_0/I_0 is the input resistance at that point. This parameter is of great functional significance because it provides a measure for the "responsiveness" of a specific region to its inputs. It is also a quantity, which unlike R_M , may be directly measured. From Ohm's law, we know that a dendritic region with a large R_{in} requires only a small input current (a small excitatory conductance change) to produce a significant voltage change locally, at the input site. Conversely, a region with small R_{in} requires a more powerful input or several inputs to generate a significant voltage change locally.

In the case of an infinite cylinder, when a steady current input is injected at some point $x = 0$, the input current must divide into two equal core currents; one half flows to the right at $x = 0$ and the other half flows to the left. Thus, from Equation 7-1

$$I_i = -\frac{1}{r_i}\frac{\partial V}{\partial x}\bigg|_{x=0} = \frac{I_0}{2} \qquad \text{Eq. 7-50}$$

From Equation 7-22, the derivative $(\partial V/\partial x)\big|_{x=0} = -V_0/\lambda$. We then get;

$$R_{in} = \frac{V_0}{I_0} = \frac{r_i\lambda}{2} = \frac{\sqrt{r_m r_i}}{2} \qquad \text{Eq. 7-51}$$

or

$$R_{in} = \left(\frac{1}{\pi}\right) d^{-3/2}\sqrt{R_M R_A} \qquad \text{Eq. 7-52}$$

For the semi-infinite cylinder, the input resistance (often represented by R_∞) will be twice this amount. Hence, in an infinitely long cylinder, the input resistance is directly proportional to the square root of the specific membrane and axoplasm resistivities, and is inversely proportional to the core diameter, raised to the 3/2 power. Consequently, thin cylinders have a larger R_{in} compared to thicker cylinders that have the same R_M and R_A values. The dependence of the input resistance on $d^{3/2}$ was utilized by Rall (1959) [4] to develop the concept of "trees equivalent to a cylinder." Rall argued that when a cylinder with diameter d_p bifurcates into two daughter branches with diameters d_1 and d_2 (and both daughter branches have the same boundary conditions at the same value of L), the branch point behaves as a continuous cable for current that flows from the parent to daughters, if

Axoplasm Resistivity

Axoplasm is the cytoplasm within the axon of a neuron (nerve cell). The electrical **resistance** of the **axoplasm**, called **axoplasmic resistance**, is one aspect of a neuron's cable properties, because it affects the rate of travel of an action potential down an axon.

$$d_p^{3/2} = d_1^{3/2} +_2^{3/2} \qquad \text{Eq. 7-53}$$

Provided that the specific properties of membrane and cytoplasm are uniform, Eq. 5.25 implies that the sum of input conductances of the two daughter branches (at the branch point) is equal to the input conductance of the parent branch at this point (impedance matching at the branch point). Thus, a branch point that obeys Eq. 5.25 is electrically equivalent to a uniform cylinder (looking from the parent into the daughters). Rall extended this concept to trees and showed that (from the soma viewpoint out to the dendrites) there is a subclass of trees that are electrically equivalent to a single cylinder whose diameter is that of the stem (near the soma)

dendrite. (See Rall (1959 [4], 1989 [24]) and Jack et al. (1975) [18]). It was surprising to find that dendrites of many neuron types (e.g., the a-motoneuron in Figure 7-1B) obey, to a first approximation, the $d^{3/2}$ rule (Equation 7-53). (See, for example, Bloomfield, Hamos and Sherman (1987) [27]). However, the dendrites of several major types of neurons (e.g., cortical and hippocampal pyramidal cells) do not obey this rule. Still, the "equivalent cylinder" model for dendritic trees allows for a simple analytical solution (Rall and Rinzel 1973, Rinzel and Rall 1974) and, indeed, it provided the main insights regarding the spread of electrical signals in passive dendritic trees, as summarized in Section 7.6.

It may be, shown that the input resistance for a finite cylinder with sealed end at $X = L$ is larger by a factor of $\coth(L)$ than that of a semi-infinite cylinder having the same membrane and axial resistance, and the same diameter. When the end at $X = L$ is clamped to rest, the input resistance is smaller than that of the semi-infinite cylinder by a factor of $\tanh(L)$ (See Rall (1989) [24] for complete derivations.) This leads to the useful result that, if the neuron and its associated dendritic tree can be approximated by an equivalent sealed end cylinder of surface area A and electrotonic length L, then

$$R_M = \frac{R_{in}}{L}\tanh(L) \qquad \text{Eq. 7-54}$$

This provides a way to estimate the specific membrane resistance R_M from the measured input resistance if A and L are known, or to estimate the dendritic surface area if R_M and L are known (Rall 1977 [23], 1989 [19]).

7.4.10. Summary of Main Results from the Cable Equation

In view of the solutions for the three representative cases, the infinite cylinder (Equation 7-22), the finite cylinder with sealed end (Equation 7-23) and the finite cylinder with end clamped to the resting potential (Equation 7-25), it is important to emphasize a few points:

1. The attenuation of steady voltage is, determined solely by the space constant l, only in the case of infinite cylinders. In this case, steady voltage attenuates *e*-fold per unit of length l. In finite cylinders, however, λ is not the sole determinant of this attenuation; the electrical length of the cylinder L and the boundary condition at the end toward which current flows (and voltage attenuates) also determine the degree of attenuation along the cylinder (Figure 7-3).
2. The "sealed end" or "open circuit" boundary condition and the "clamped to rest" termination are two extremes. In dendritic trees, short dendritic segments terminate by a sub-tree that imposes "leaky" boundary conditions at the segment's ends. The size of the sub-tree and its electrical properties determine how leaky the conditions are at the boundaries of any given segment. In general, when a large tree is, connected at the end of the dendritic segment, the leaky boundary condition at this

end approaches the "clamped to rest" condition. When current flows in the direction of such a leaky end, voltage attenuates steeply along this segment. In contrast, when the sub-tree is very small the leaky boundary conditions approach the "sealed-end" condition and a very shallow attenuation is, expected towards such an end (Figure 7-6A). Rall (1959) [4] showed how to compute analytically the various boundary conditions at any point in a passive tree with arbitrary branching and specified *RM*, *RA* and (for the transient case) C_M values (See Jack et al. (1975) [18], Segev et al. (1989) [6]).

3. An important consequence of this dependence of voltage attenuation on the boundary conditions in dendritic trees is that this attenuation is asymmetric in the central (from dendrites to soma) vs. the peripheral (away from soma) directions. In general, because the boundary conditions are more "leaky" in the central direction, voltage attenuation in this direction is steeper than in the peripheral one. Figure 7-6A illustrates this important point very clearly.
4. Dendritic trees can be, approximated (electrically), to a first degree, by a single (finite) cylinder. Therefore, analysis of the behavior of voltage in such cylinders provides important insights into the behavior of voltage in dendritic trees.
5. By peeling the slowest (τ_0) and the first equalizing time constant (τ_1) from somatic voltage transients, the electrical length L of the dendritic tree could be, estimated, assuming that the tree is equivalent to a single cylinder (Equation 7-49). Indeed, utilizing this peeling method for many neuron types we know that, depending on the neuron.

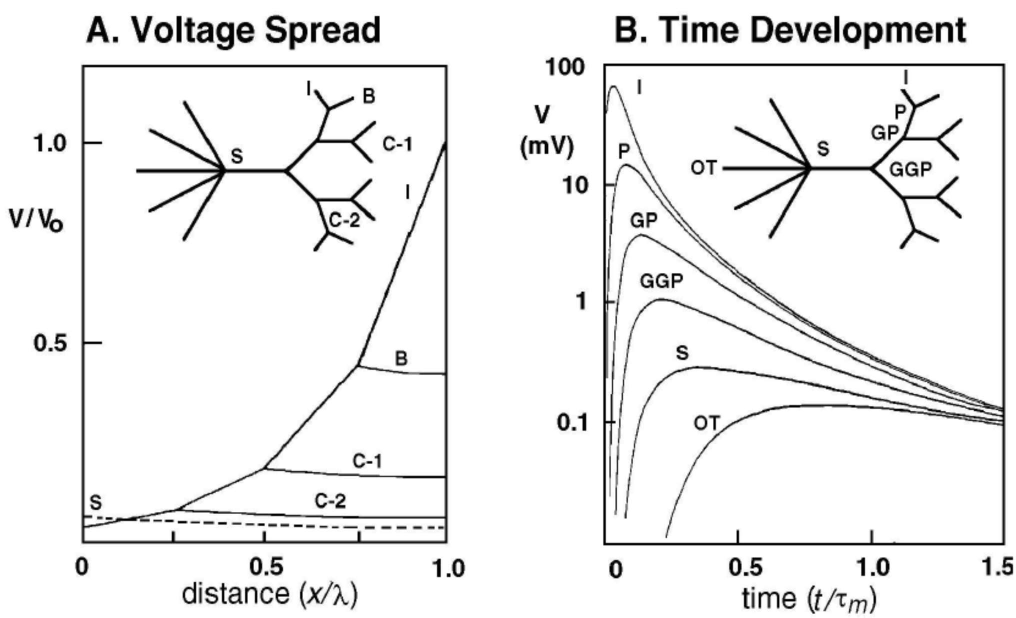


Figure 7-6. The voltage spread in passive dendritic trees is asymmetrical (A); its time-course changes (is, broadened) and the peak is, delayed as it propagates away from the input site (B). Solid curve in (A) shows the steady-state voltage computed for current input to terminal branch I.

Large attenuation is, expected in the input branch whereas much smaller attenuation exists in its identical sibling branch B. The dashed line corresponds to the same current when applied to the soma. Note the small difference, at the soma, between the solid curve and the dashed curve, indicating the negligible "cost" of placing this input at the distal branch rather than at the soma. (Data is, replotted from Rall and Rinzel (1973)) [27]. In (B), a brief transient current is applied to terminal branch I and the resultant voltage at the indicated points is shown on a logarithmic scale. Note the marked attenuation of the peak voltage (by several hundredfold) from the input site to the soma and the broadening of the transient as it spreads away from the input site. (Data is, replotted from Rinzel and Rall (1974) [28]). Dendritic terminals have sealed ends in both (A) and (B).

7.5. COMPARTMENTAL MODELING APPROACH

The compartmental modeling approach complements cable theory by overcoming the assumption that the membrane is passive and the input is current (Rall 1964) [5]. Mathematically, the compartmental approach is a finite-difference (discrete) approximation to the (nonlinear) cable equation. It replaces the continuous cable equation by a set, or a matrix, of ordinary differential equations and, typically, numerical methods are, employed to solve this system (which can include thousands of compartments and thus thousands of equations) for each time step. Conceptually, in the compartmental model dendritic segments that are electrically short are, assumed to be, isopotential and are lumped into a single R-C (either passive or active) membrane compartment (Figure 7-7). Compartments are, connected to each other via a longitudinal resistivity according to the topology of the tree. Hence, differences in physical properties (e.g., diameter, membrane properties, etc.) and differences in potential occur between compartments rather than within them. It can be, shown that when the dendritic tree is, divided into sufficiently small segments (compartments) the solution of the compartmental model converges to that of the continuous cable model. A compartment can represent a patch of membrane with a variety of voltage-gated (excitable) and synaptic (time-varying) channels. A review of this very popular modeling approach can be, found in Segev, Fleshman and Burke (1989) [29].

In the cable representation (B), the voltage can be, computed at any point in the tree by using the continuous cable equation and the appropriate boundary conditions imposed by the tree.

An analytical solution can be, obtained for any current input in passive trees of arbitrary complexity with known dimensions and known specific membrane resistance and capacitance (R_M, C_M) and specific cytoplasm (axial) resistance (R_A). In the compartmental representation, the tree is, discretized into a set of interconnected R-C compartments. Each is a lumped representation of the membrane properties of a sufficiently small dendritic segment. Compartments are, connected via axial cytoplasmic resistances. In this approach, the voltage can be, computed at each compartment for any (nonlinear) input and for voltage and time-dependent membrane properties.

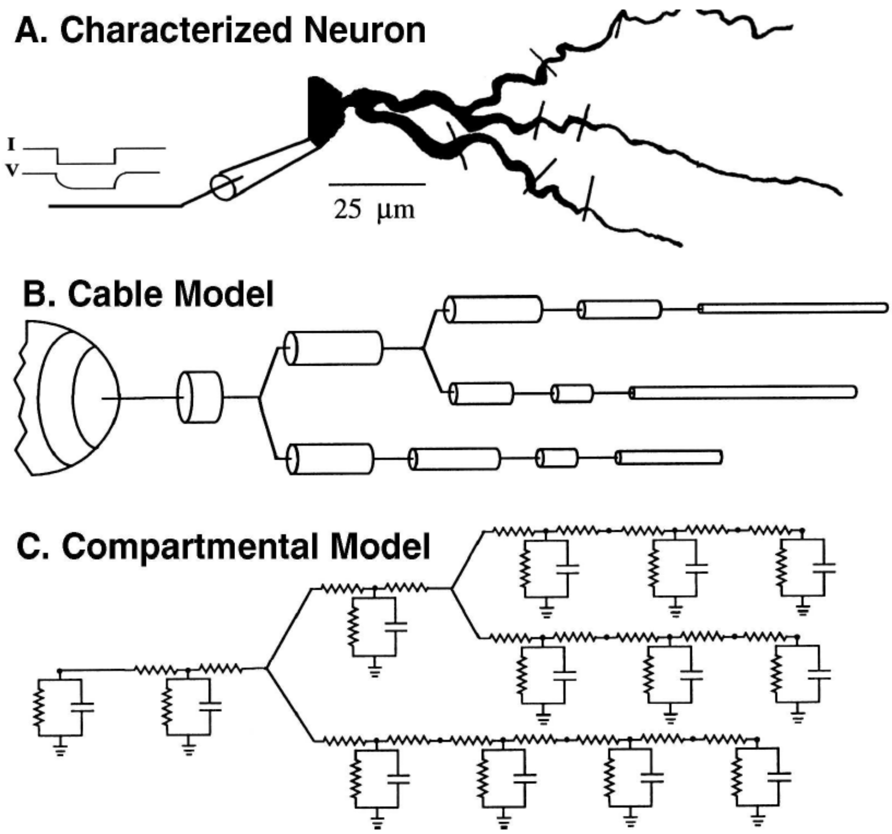


Figure 7-7. Dendrites (A) are, modeled either as a set of cylindrical membrane cables (B) or as a set of discrete isopotential R-C compartments (C).

As an example, consider a section of a uniform cylinder, divided into a number of identical compartments, each of length l. If we introduce an additional current I_j to represent the flow of ions from the jth compartment through active (nonlinear synaptic and/or voltage-gated) channels, we can write Equation 7-3 as

$$\frac{l^2}{R_a}\frac{\partial^2 V_j}{\partial x^2} = C_m \frac{\partial V_j}{\partial t} + \frac{V_j}{R_m} + I_j \qquad \text{Eq. 7-55}$$

Here, V_j represents the voltage in the jth compartment, and we have used Equations. 7-4 -7 -6 to give the actual values of the resistances and capacitances (in ohms and farads) of this compartment, instead of the values for a unit length. Note that now R_m represents the membrane resistance at rest, before the membrane potential (and membrane resistance) is changed due to the current I_j. Also, note that V_j appears in the expression for I_j. For

example, in the case of synaptic input to compartment j, $I_j = g(t)(V_j - E_{syn})$. Here, $g(t)$ and E_{syn} are synaptic conductance and reversal potential, respectively [15].

It can be shown by use of Taylor's series that for small values of *l*, the left hand side of Equation 7-55 can be expressed in terms of differences between the value of V_j and the values in the adjacent compartments, V_{j-1} and V_{j+1}. In this approximation, the cable equation becomes

$$\frac{V_{j+1} - V_j + V_{j-1}}{R_a} = C_m \frac{dV_j}{dt} + \frac{V_j}{R_m} + I_j \qquad \text{Eq. 7-56}$$

For the general case of a dendritic cable of non-uniform diameter (in which R_m, R_a and C_m may vary among compartments), we obtain the result given earlier in Equation 6-2 and discussed in Section 6.7.5. This equation can easily be, extended to include a branch structure. For a tree represented by N compartments we get N coupled equations of the form of Equation 7-56.

They should be solved simultaneously to obtain V_j, for $j = 1, 2, \cdots, N$ at each time step Δt.

7.5.1. Compartmental Modes

We have seen that analytical solutions can be, given for the voltage along a passive cable with uniform geometrical and electrical properties. If we want to apply the above results in order to describe the membrane potential along the dendritic tree of a neuron we face several problems. Even if we neglect `active' conductances formed by non-linear ion channels a dendritic tree is at most locally equivalent to a uniform cable. Numerous bifurcations and variations in diameter and electrical properties along the dendrite render it difficult to find a solution for the membrane potential analytically (Abbott et al., 1991).

Numerical treatment of partial differential equations such as the cable equation requires a discretization of the spatial variable. Hence, all derivatives with respect to spatial variables are, approximated by the corresponding quotient of differences. Essentially, we are, led back to the discretized model of Figure 7-2, which has been, used as the starting point for the derivation of the cable equation. After the discretization, we have a large system of ordinary differential equations for the membrane potential at the chosen discretization points as a function of time. This system of ordinary differential equations can be, treated by standard numerical methods. In Figure 7-2, we can see, as part of a dendrite and the corresponding circuit diagram. Longitudinal and transversal resistors are denoted by R_L and R_T,

respectively. The electrical capacity of each small piece of dendrite is, symbolized by capacitors C.

In order to solve for the membrane potential of a complex dendritic tree numerically, compartmental models are used that are the result of the above, mentioned discretization (Bower and Beeman, 1995 [15]; Yamada et al., 1989 [30]; Ekeberg et al., 1991 [31]). The dendritic tree is, divided into small cylindric compartments with an approximatively uniform membrane potential. Each compartment is, characterized by its capacity and transversal conductivity. The longitudinal resistance that is determined by their geometrical properties (Figure 7-8) couples adjacent compartments.

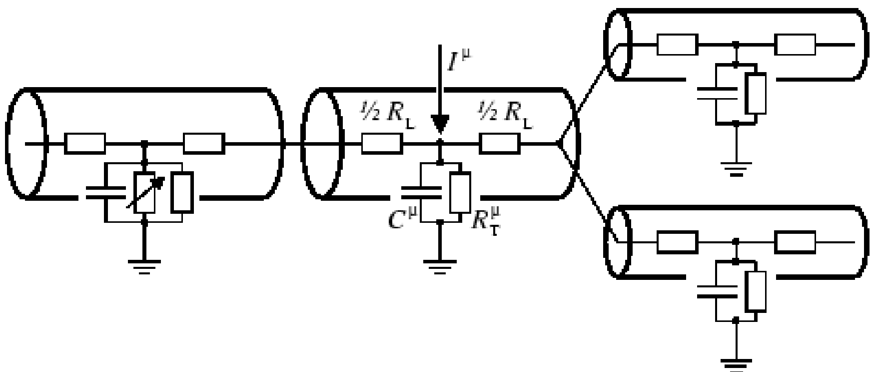


Figure 7-8. Multi-compartment neuron model. Dendritic compartments with membrane capacitance C^{μ} and transversal resistance R_T^{μ} are coupled by a longitudinal resistance $(r^{\nu})^{\mu} = (R_T^{\nu} + R_T^{\mu})/2$. External input to compartment μ is denoted by I^{μ}. Some or all compartments may also contain nonlinear ion channels (variable resistor in leftmost compartment).

Once numerical methods are, used to solve for the membrane potential along the dendritic tree, some or all compartments can be equipped with nonlinear ion channels as well. In this way, effects of nonlinear integration of synaptic input can be studied (Mel, 1994) [32]. Apart from practical problems that arise from a growing complexity of the underlying differential equations, conceptual problems are, related to a drastically increasing number of free parameters. The more so, since almost no experimental data regarding the distribution of any specific type of ion channel along the dendritic tree is available. To avoid these problems, all nonlinear ion channels responsible for generating spikes are usually lumped together at the soma and the dendritic tree is, treated as a passive cable. For a review of the compartmental approach, we refer the reader to the book of Bower and Beeman (Bower and Beeman, 1995) [15]. In the following, we illustrate the compartmental approach by a model of a cerebellar granule cell.

7.5.2. A Multi-Compartment Model of Cerebellar Granule Cells

As an example for a realistic neuron model, we discuss a model for cerebellar granule cells in turtle developed by Gabbiani and coworkers (Gabbiani et al., 1994) [33]. Granule

cells are extremely numerous tiny neurons located in the lowest layer of the cerebellar cortex. These neurons are particularly interesting because they form the sole type of excitatory neuron of the whole cerebellar cortex (Ito, 1984) [34].

Figure 7-8, shows a schematic representation of the granule cell model. It consists of a spherical soma and four cylindrical dendrites that are, made up of two compartments each. There is a third compartment at the end of each dendrite, the dendritic bulb, which contains synapses with mossy fibers and Golgi cells.

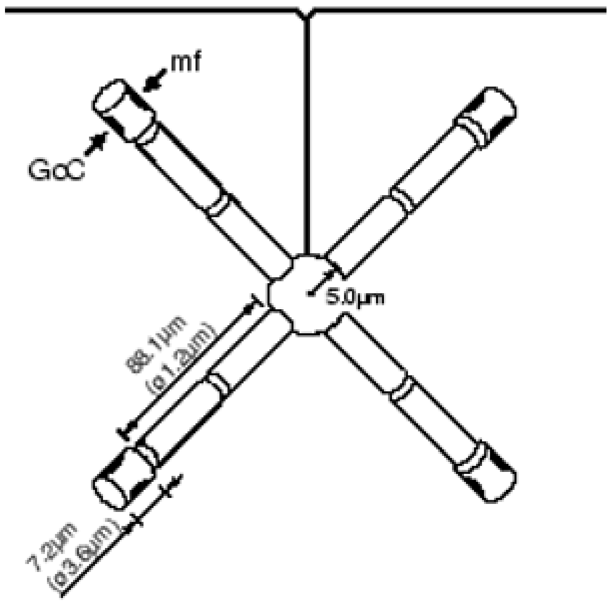


Figure 7-8. Schematic Representation of the Granule Cell Model (not to scale).

The model consists of a spherical soma (radius 5.0 µm) and four cylindrical dendrites (diameter 1.2 µm, length 88.1 µm) made up of two compartments each. There is a third compartment at the end of each dendrite, the dendritic bulb, which contains synapses with mossy fibers (mf) and Golgi cells (GoC). The active ion channels are located at the soma. The dendrites are passive. The axon of the granule cell, which rises vertically towards the surface of the cerebellar cortex before it undergoes a T -shaped bifurcation, is not included in the model.

One of the major problems with multi-compartment models is the fact that the *spatial distribution* of ion channels along the surface of the neuron is almost completely unknown. In the present model, it is therefore, assumed for the sake of simplicity that all active ion channels are concentrated at the soma. The dendrites, on the other hand, are, described as a passive cable.

The granule cell model contains a fast Sodium current I_{Na} and a calcium-activated Potassium current $I_{\text{K(Ca)}}$ that provide a major contribution for generating action potentials.

7.6. MAIN INSIGHT FOR PASSIVE DENDRITES WITH SYNAPSES

Here we summarize the main insights regarding the input-output properties of passive dendrites, gained from modeling and experimental studies on dendrites during the last forty years.

1) Dendritic trees are electrically distributed (rather than isopotential) elements. Consequently, voltage gradients exist over the tree when synaptic inputs are, applied locally. Because of inherent non-symmetric boundary conditions in dendritic trees, voltage attenuates much more severely in the dendrites-to-soma direction than in the opposite direction (Figure 7-6A). In other words, from the *soma* viewpoint, dendrites are electrically rather compact (average cable length L of 0.3–2 λ). From the *dendrite* (synaptic) viewpoint, however, the tree is electrically far from being compact. A corollary of this asymmetry is that, as seen in Figure 7-6A, short side branches (in particular, dendritic spines) are essentially isopotential with the parent dendrite when current flows from the parent into these side branches (spines) but a large voltage attenuate.
2) The large voltage attenuation from dendrites to soma, which may be a few hundredfold for brief synaptic inputs at distal sites, implies that many several tens of excitatory inputs should be activated within the integration time window τ_m . in order to build up depolarization of 10–20 *mV* and reach threshold for firing of spikes at the soma and axon (e.g., Otmakhov, Shirke and Malinow 1993 [36], Barbour 1993 [37]). The large local depolarization is, expected at the dendrites, together with the marked attenuation in the tree imply that the tree can be, functionally, separated into many, almost independent, subunits (Koch, Poggio and Torre 1982) [37].
3) Although severely attenuated in peak values, the attenuation of the area of transient potentials (as well as the attenuation of charge) is relatively small. The attenuation of the area is identical to the attenuation of the steady voltage, independently of the transient shape (Rinzel and Rall 1974) [38]. Comparing the steady-state somatic voltage that results from distal dendritic input to the soma voltage when the same input is, applied directly to the soma (dashed line in Figure 7-6A) highlights this point. Thus, the "cost" (in term of area or charge) of placing the synapse at the dendrites rather than at the soma is quite small.
4) Linear system theory implies an interesting reciprocity in passive trees. The voltage at some location X_j resulting from transient current input at point *xi* is identical to the voltage transient measured at X_i when the same current input is, applied at X_j (Koch et al. 1982) [37]. Because the input resistance is typically larger at thin distal dendrites and in spines than at proximal dendrites (and the soma), the same current produces a larger voltage response at distal dendritic arbors. Thus, the reciprocity theorem implies that the *attenuation* of voltage from these sites to the soma is steeper than in the opposite direction (i.e., asymmetrical attenuation). Reciprocity also holds for the total signal delay between any given points in the dendritic tree (Agmon-Snir and Segev 1993) [39].

5) Synaptic potentials are, delayed, and they become significantly broader, as they spread away from the input site (Figure 7-6B). The large sink provided by the tree at distal arbors implies that, locally, synaptic potentials are very brief. At the soma level, however, the time-course of the synaptic potentials is primarily governed by τ_m. This change in width of the synaptic potential implies multiple time windows for synaptic integration in the tree (Agmon-Snir and Segev 1993) [39].
6) Excluding very distal inputs, the cost (in terms of delay) that results from dendritic propagation time (i.e., the net dendritic delay) is small compared to the relevant time window (τ_m) for somatic integration. Thus, for the majority of synapses, the significant time window for *somatic* integration remains τ_m (Agmon-Snir and Segev 1993) [39].
7) Because of the inherent conductance change (shunt) associated with synaptic inputs, synaptic potentials sum nonlinearly (less than linear) with each other (Chapter 6). This local conductance change of the membrane is better "felt" by electrically adjacent synapses than by more remote (electrically decoupled) synapses. Consequently, in passive trees, spatially distributed excitatory inputs sum more linearly (produce more charge) than do spatially clustered synapses (Rall 1964) [5].
8) Inhibitory synapses (whose conductance change is associated with a battery near the resting potential) are more effective when located on the path between the excitatory input and the "target" point (soma) than when placed distal to the excitatory input. Thus, when strategically placed, inhibitory inputs can specifically veto parts of the dendritic tree and not others (Rall 1964 [5], Koch et al. 1982 [37], Jack et al. 1975 [18]).
9) Because of dendritic delay, the somatic depolarization that results from activation of excitatory inputs at the dendrites is very sensitive to the temporal sequence of the synaptic activation. It is largest when the synaptic activation starts at distal dendritic sites and progresses proximally. Activation of the same synapses in the reverse order in time will produce smaller somatic depolarization. Thus, the output of neurons with dendrites is inherently directionally selective (Rall 1964) [5].
10) Because the synaptic input changes the membrane conductance, it effectively alters the cable properties (electrotonic length, input resistance, time constant, etc.) of the postsynaptic cell. This activity can reduce the time constant by a factor of 10 (Bernander et al. 1991 [40], Rapp et al. 1992) [41]. Thus, spontaneous (background) synaptic activity dynamically changes the computational (input-output) capabilities of the neuron.

7.7. Biophysics of Excitable Dendrites

A growing body of experimental evidence in recent years has clearly demonstrated that the membrane of many types of dendrites is endowed with voltage-gated (nonlinear) ion channels, including the NMDA channels as well as voltage-activated inward (Ca^{+2} and Na^{+}) and outward K^{+}) conductances (e.g., Stuart and Sakmann 1994 [42], Laurent 1993 [43],

McKenna et al. 1992 [45], Wilson 1992) [45]. These channels are responsible for a variety of sub-threshold electrical nonlinearities and, under favorable conditions; they can generate full-blown action potentials. The use of voltage- and ion-dependent dyes as well as intracellular and patch-clamp recordings from dendrites suggested that, in contrast to axonal trees, the regenerative phenomenon from input into excitable dendrites tends to spread only locally.

This makes functional sense since; otherwise, the dendritic tree would be essentially no different from the axon, implementing a simple all-or-none operation. However, because of the asymmetric spread of voltages within the dendritic tree (Figure 7-6A) and because of inhomogeneous distribution of excitable channels in dendrites, spikes can propagate more readily back from the soma to the dendrites (Stuart and Sakmann 1994) [42]. Unfortunately, we still lack information regarding the distribution, the voltage-dependence, and the kinetics of excitable channels in dendrites and most of the results of this section are primarily, based on theoretical predictions.

NMDA Receptor Definition

The N-methyl-D-aspartate receptor (also known as the NMDA receptor or NMDAR), is a glutamate receptor and ion channel protein found in nerve cells. It is activated when glutamate and glycine (or D-serine) bind to it, and when activated it allows positively charged ions to flow through the cell membrane. The NMDA receptor is very important for controlling synaptic plasticity and memory function

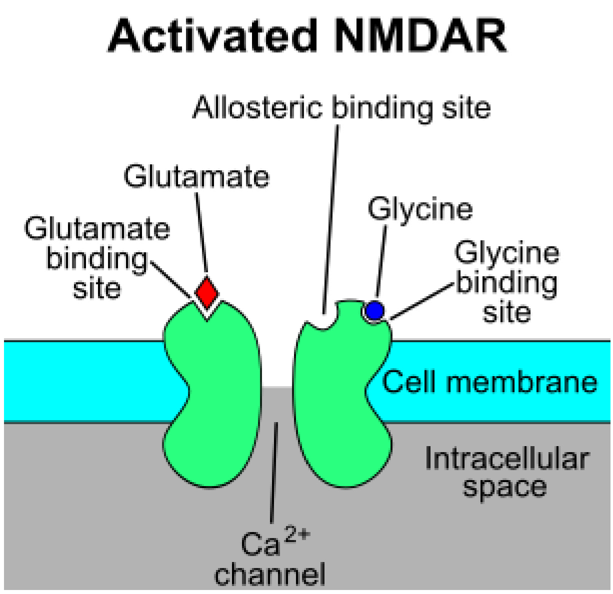


Figure 7-9. Stylized Depiction of an Activated NMDAR.

Stylized depiction of an activated NMDAR is, illustrated in the above figure. Glutamate is in the glutamate-binding site and glycine is in the glycine-binding site. The allosteric site, which modulates receptor function when bound to a ligand, is not, occupied. NMDARs require the binding of two molecules of glutamate or aspartate and two of glycine.

What is the electrical behavior to be, expected from dendrites with voltage-gated membrane ion channels? First, we note that the presence of voltage-gated channels in dendrites does not automatically imply that these channels participate in the electrical activity of the tree under all conditions. The large conductance load imposed by the tree effectively increases the activation threshold of these channels for local synaptic potentials (Rall and Segev 1987). These channels will be more readily activated under favorable conditions such as in regions with high densities of excitable channels (as in the initial segment of the axon), when the excitable channels have fast activation kinetics, or when the input is distributed (not localized). When activated, these channels can modulate the input-output properties of the neuron. For example, they can *amplify* the excitatory synaptic current and, for channels that carry inward current, the regenerative activity can spread ("chain reaction") and indirectly activate nearby dendritic regions that will further enhance the excitatory synaptic inputs (Rall and Segev 1987). Consequently, distal excitatory synaptic inputs may be less attenuated and, thus, affect more strongly the neuron output than would be expected in the passive case. In general, because of the asymmetry of voltage attenuation in dendritic trees (Figure 7-6A), regenerative activity in dendrites will spread more securely in the centrifugal (soma-to-dendrites) direction than in the centripetal direction. This effect may be, explored in the simulation (*traub91*) of the CA3 pyramidal cell described in Chapter 7 of Bower and Beeman [15]. In addition to modulating the strength of the synaptic current, the kinetics of excitable channels may also play an important role in modulating the *speed* of electrical interaction in the dendritic tree.

Another consequence of dendritic nonlinearity was, discussed by Mel (1993)46. Unlike the passive case where synaptic saturation implies loss in synaptic efficacy when the synapses are spatially clustered (see number 7 above), in the excitable situation (including the case of the voltage- and transmitter-gated NMDA receptors) a certain degree of input clustering implies more charge transfer to the soma (due to the extra active inward current). In this case, the output at the axon depends sensitively on the size (and site) of the "clusteron" and this may serve as a mechanism for implementing a multi-dimensional discrimination task of input patterns via multiplication-like operation.

Recently, Wilson (1992) [45] put the possibility that active dendritic currents (both inward and outward) may serve as a mechanism for synaptic gain control forward in the context of neostriatal neurons and by Laurent (1993) for the axonless none spiking interneurons of the locust. The principal idea is that, as, a result of active currents, the integrative capabilities of the neuron (e.g., its input resistance and electrotonic length) are dynamically controlled by the membrane potential; thereby the neuron output depends on its state (membrane potential). Active currents (e.g., outward K^+ current) can act to counterbalance excitatory synaptic inputs (negative feedback) and thus stabilize the input-output characteristics of the neuron. Conversely, at other voltage regimes, active currents might effectively increase the input resistance and reduce the electrotonic distance between synapses (positive feedback) with the consequence of nonlinearly boosting a specific group of coactive excitatory synapses.

7.8. Computational Function of Dendrites

It seems appropriate to conclude this chapter by asking what kind of computations could be, performed by a neuron with dendrites that could not be, carried out with just a formless point neuron. Several answers have already been, discussed; here the major ones are succinctly, highlighted:

1) Neurons with dendrites can compute the direction of motion (Rall 1964, Koch et al. 1982).
2) Neurons with dendrites can simultaneously function on multiple time windows. For local dendritic computations (e.g., triggering local dendritic spikes, triggering local plastic processes) distal arbors act more as coincidence detectors, whereas the soma acts more as an integrator when brief synaptic inputs (i.e., non-NMDA and GABAA) are involved (Agmon-Snir and Segev 1993).
3) Neurons with dendrites can implement a multi-dimensional classification task (Mel 1993).
4) Neurons with dendrites can function as many, almost independent, functional subunits. Each unit can implement a rich repertoire of logical operations (Koch et al. 1982 [37], Rall and Segev 1987 [47]) as well as other local computations (e.g., local synaptic plasticity) and they can function as semi-autonomous input-output elements (e.g., via dendrodendritic synapses).
5) Neurons with slow ion currents in the dendrites that are partially, decoupled from fast spike-generating currents at the soma/axon hillock can produce a large repertoire of frequency patterns. By modulating the degree of electrical coupling between the dendrites and the soma (e.g., by inhibition) the same input can produce regular high frequency spiking as well as bursting — as thought to occur in experimental and theoretical models of epileptic seizures (Pinsky and Rinzel 1994) [48].

7.9. Summary

"Real" neurons are extremely complex biophysical and biochemical entities. Before designing a model, it is therefore necessary to develop an intuition for what is important and what can be safely, neglected. The Hodgkin-Huxley model describes the generation of action potentials on the level of ion channels and ion current flow. It is the starting point for detailed neuron models, which in general include more than the three types of currents considered by Hodgkin and Huxley.

Electro-physiologists have described an overwhelming richness of different ion channels. The set of ion channels is different from one neuron to the next. The precise channel configuration in each individual neuron determines a good deal of its overall electrical properties. Synapses are usually, modeled as specific ion channels that open for a certain time after presynaptic spike arrival.

The geometry of the neuron can play an important role in synaptic integration because the effect of synaptic input on the somatic membrane potential depends on the location of the synapses on the dendritic tree. Though some analytic results can be, obtained for passive

dendrites, it is usually necessary to resort to numerical methods and multi-compartment models in order to account for complex geometry and active ion channels.

REFERENCES

[1] Rapp, M., Yarom, Y. and Segev, I. (1992), The impact of parallel fiber background activity on the cable properties of cerebellar Purkinje cells, *Neural Computation* 4: 518–533.

[2] Burke R. Spinal motoneuron *Neuroscience in the 21st Century: From Basic to Clinical*. 1027-1062.

[3] Wilson, C. J. (1992). Dendritic morphology, inward rectification and the functional properties of Neostriatal neurons, *in* T. McKenna, J. Davis and S. Zornetzer (eds), *Single Neuron Computation*, Academic Press, San Diego, pp. 141–172.

[4] Rall, W. (1959), Branching dendritic trees and motoneuron membrane resistivity, *Exp. Neurol.* 1: 491–527.

[5] Rall, W. (1964), Theoretical significance of dendritic trees for neuronal input-output relations, *in* R. Reiss (ed.), *Neuronal Theory and Modeling*, Stanford University Press, Stanford, CA, pp. 73–97.

[6] Segev, I., Rinzel, J. and Shepherd, G. H. (Eds) (1995), *The Theoretical Foundation of Dendritic Function: Selected Papers by Wilfrid Rall with Commentaries*, MIT Press, Cambridge, MA.

[7] White, E. L. (1989). *Cortical circuits: Synaptic organization of the cerebral cortex — Structure, function and theory*, Birkhȧuser, Boston.

[8] Shepherd, G. M. (1990), *The Synaptic Organization of the Brain*, third edn, Oxford University Press, New York.

[9] Segev, I. (1995). Denritic processing, *in* M. A. Arbib (ed.), *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, MA.

[10] Segev, I. and Rall, W. (1988), Computational study of an excitable dendritic spine, *J. Neurophsiol.* 60: 499–523.

[11] Koch, C. and Zador, T. (1993), The function of dendritic spines: Devices subserving biochemical rather than electrical compartmentalization, *J. Neuroscience* 13: 413–422.

[12] Koch, C. and Zador, T. (1993), The function of dendritic spines: Devices subserving biochemical rather than electrical compartmentalization, *J. Neuroscience* 13: 413–422.

[13] Stuart, G. J. and Sakmann, B. (1994), Active propagation of somatic action potentials into neocortical pyramidal cell dendrites, *Nature* 367: 69–72.

[14] Shepherd, G. M. (1990), *The Synaptic Organization of the Brain*, third edn, Oxford University Press, New York.

[15] Bower, J. and Beeman, D. The Book of GENESIS, Exploring Realistic Neural Models with the General Neural Simulation System, 2nd edition, 1997, Springer Publishing Company.

[16] Zohuri, B., Dimensional Analysis and Self-Similarity Methods for Engineers and Scientists Apr 16, 2015.

[17] Rall, W. (1989), Cable theory for dendritic neurons, *in* C. Koch and I. Segev (eds), *Methods in Neuronal Modeling*, MIT Press, Cambridge, MA, chapter 2, pp. 9–62.

[18] Jack, J. J. B., Noble, D. and Tsien, R. W. (1975), *Electric Current Flow in Excitable cells*, Calderon Press, Oxford.
[19] Rall, W. (1989), Cable theory for dendritic neurons, in C. Koch and I. Segev (eds), Methods in Neuronal Modeling, MIT Press, Cambridge, MA, chapter 2, pp. 9–62.
[20] Jackson, J. D. *Classical Electrodynamics*. Wiley Published by John, 1962.
[21] Abbott, L.F. (1991), Realistic synaptic inputs for model neural networks. *Network*, 2:245-258.
[22] Rall, W. (1969) Time constant and electrotonic length of membrane cylinders and neurons, *Biophys. J.* 9: 1483–1508.
[23] Rall, W (1977), Cable theory for neurons, in E. R. Kandel, J. M. Brookhardt and V. B. Mountcastle (eds), Handbook of Physiology: The Nervous System, Vol. 1, Williams and Wilkins, Baltimore, chapter 3, pp. 39–98.
[24] Rall, W (1967). Distinguishing theoretical synaptic potentials computed for different somadendritic distribution of synaptic inputs, *J. Neurophysiol.* 30: 1138–1168.
[25] Ranck, J. B. (1973), Studies on single neurons in dorsal hippocampal formation and septum in unrestrained rats. I. behavioral correlates and firing repertoires, *Exp. Neurol.* 41: 462–531.
[26] Bloomfield, S. A., Hamos, J. E. and Sherman, S. M. (1987). Passive cable properties and morphological correlates of neurones in the lateral geniculate nucleus of the cat, *J. Physiol. (London)* 383: 653–692.
[27] Rall, W. and Rinzel, J. (1973). Branch input resistance and steady state attenuation for input to one branch of a dendritic neuron model, *Biophys. J.* 13: 648–688.
[28] Rinzel, J. and Rall, W. (1974), Transient response in a dendritic neuron model for current injected at one branch, *Biophys. J.* 14: 759–790.
[29] Segev, I., Fleshman, J. W. and Burke, R. E. (1989), Compartmental models of complex neurons, *in* C. Koch and I. Segev (eds), *Methods in Neuronal Modeling*, MIT Press, Cambridge, MA, chapter 3, pp. 63–96.
[30] Yamada, W. M., Koch, C., and Adams, P. R. (1989), Multiple channels and calcium dynamics. In Koch, C. and Segev, I., editors, *Methods in neuronal modeling: From synapses to networks*, chapter 4. MIT Press, Cambridge, MA.
[31] Ekeberg, O., Wallen, P., Lansner, A., Traven, H., Brodin, L., and Grillner, S. (1991). A computer based model for realistic simulations of neural networks. *Biol. Cybern.*, 65:81-90.
[32] Mel, B. W. (1994), Information processing in dendritic trees. *Neural Comput.*, 6(1031-1085).
[33] Gabbiani, F., Midtgaard, J., and Knoepfl, T. (1994), Synaptic integration in a model of cerebellar granule cells. *J. Neurophysiol.*, 72:999-1009. Corrigenda have been published in J. Neurophysiol. (1996) 75(6), without covering, however, all typing errors.
[34] Ito, M. (1984). *The Cerebellum and Neural Control*. Raven Press, New York.
[35] Otmakhov, N., Shirke, A. M. and Malinow, R. (1993), Measuring the impact of probabilistic transmission on neuronal output, *Neuron* 10: 1101–1111.
[36] Barbour, B. (1993). Synaptic currents evoked in Purkinje cells by stimulating individual granule cells, *Neuron* 11: 759–769.
[37] Koch, C., Poggio, T. and Torre, V. (1982), Retinal ganglion cells: a functional interpretation of dendritic morphology, *Phil. Trans. R. Soc. Lond. (Biol.)* 298: 227–264.

[38] Rinzel, J. and Rall, W. (1974). Transient response in a dendritic neuron model for current injected at one branch, *Biophys. J.* 14: 759–790.

[39] Agmon-Snir, H. and Segev, I. (1993), Signal delay and input synchronization in passive dendritic structures, *J. Neurophysiol.* 70: 2066–2085.

[40] Bernander, O., Douglas, R. D., Martin, K. A. and Koch, C. (1991). Synaptic background activity influences spatiotemporal integration in single pyramidal cells, *Proc. Natl. Acad. Sci. (USA)* 88: 11569–11573.

[41] Rapp, M., Yarom, Y. and Segev, I. (1992), The impact of parallel fiber background activity on the cable properties of cerebellar Purkinje cells, *Neural Computation* 4: 518–533.

[42] Stuart, G. J. and Sakmann, B. (1994), Active propagation of somatic action potentials into neocortical pyramidal cell dendrites, *Nature* 367: 69–72.

[43] Laurent, G. (1993). A dendritic gain control mechanism in axonless neurons of the locust, *Schistocerca americana*, *J. Physiol. (London)* 470: 45–54.

[44] McKenna, T., Davis, J. and Zornetzer, S. F. (eds) (1992). *Single Neuron Computation*, Academic Press, San Diego.

[45] Wilson, C. J. (1992). Dendritic morphology, inward rectification and the functional properties of neostriatal neurons, *in* T. McKenna, J. Davis and S. Zornetzer (eds), *Single Neuron Computation*, Academic Press, San Diego, pp. 141–172 Mel, W. B. (1993), Synaptic integration in an excitable dendritic trees, *J. Neurophys.* 70: 1086–1101.

[46] Laurent, G. (1993). A dendritic gain control mechanism in axonless neurons of the locust, *Schistocerca americana*, *J. Physiol. (London)* 470: 45–54.

[47] Rall, W. and Segev, I. (1987), Functional possibilities for synapses on dendrites and on dendritic spines, *in* G. M. Edelman, E. E. Gall and W. M. Cowan (eds), *Synaptic Function*, Wiley, New York, pp. 605–636.

[48] Pinski, P. F. and Rinzel, J. (1994). Intrinsic and network rhythmogenesis in a reduced Traub model for CA3 neurons, *J. Comput. Neurosci.* 1: 39–60.

Chapter 8

DYNAMICS OF CEREBRAL CORTICAL NETWORKS

Goal-directed behavior requires the flexible transformation of sensory evidence about our environment into motor actions. Studies of perceptual decision making have shown that this transformation is, distributed across several widely separated brain regions. Yet, little is known about how decision making emerges from the dynamic interactions among these regions. Here, we review a series of studies, in which we characterized the cortical network interactions underlying a perceptual decision process in the human brain.

Previous chapters in this volume have considered detailed models of single cells and small networks of cells. In this chapter, we consider a large-scale multicellular model of the mammalian olfactory cortex. The simulation consists of three distinct neuronal populations of 135 cells each for a total of 405 interconnected neurons. With this simulation, we will explore the possible physiological basis for experimentally recorded electroencephalographic patterns in this cortex.

8.1. INTRODUCTION

When constructing a model of any neural system, one must always strike a balance between biological realism and computational efficiency. This is especially true in the case of network models.

The real Piriform Cortex or Pyriform Cortex of a rat, for example, contains approximately 106 neurons (Haberly 1990) [1]. Even when the complexity of individual neurons is reduced, it is still not possible to simulate all the neurons found in this network. Accordingly, the modeler is always, faced with determining the level of detail necessary to explore and illuminate the particular physiological and computational properties of these networks.

An important question then becomes; how do the physiological properties or computational capabilities of a cortical network scale with the number of neurons? In this chapter, we demonstrate how at least a rough understanding of network behavior can be, obtained with a quasi-realistic model of cerebral cortex. First, however, we briefly introduce the piriform cortex.

8.2. THE PIRIFORM CORTEX OR PYRIFORM CORTEX

The piriform cortex is the primary olfactory cortical area in the mammalian brain. For the last several years, we have been constructing realistic models of this network (Wilson and Bower 1992) [2] with the ultimate objective of understanding its role in olfactory object recognition (Hasselmo and Bower 1993 [3], Bower 1995 [4]). One motivation for this work is our assumption that this cortex computationally represents a kind of associative memory (Haberly 1985 [5], Haberly and Bower 1989 [6]).

Piriform cortex receives its afferent input from the olfactory bulb which itself receives input directly from the nasal epithelium where the olfactory receptors are located. Thus, piriform cortex is quite close to the sensory periphery, and unlike other sensory cortical areas, it does not receive its afferent input through the thalamus. Piriform cortex sends its primary projections to the entorhinal cortex but also connects to the thalamus, olfactory tubercle, superior colliculus, peri-amygdaloid, and has strong reciprocal connections with the olfactory bulb (Haberly 1990) [1] Figure 9-3 illustrates the connections between piriform cortex and related areas. Major pathways are, indicated by thick arrows. In the model, only the olfactory bulb to piriform cortex connection is included. The feedback pathway is not incorporated.

Piriform cortex is, believed to be phylogenetically older than other sensory cortical areas and is commonly, referred to as paleocortex. It also has a particularly well defined and somewhat simpler anatomical organization than neocortical regions (Haberly 1985) [5]. For example, it has only three layers instead of the six, normally found in neocortex. In this sense, it is similar to the hippocampus, which also has a trilaminar structure. There areconsiderable physiological data available on its neurons, their interactions, and on network level responses (Haberly 1990) [1]. The detailed internal structure of the piriform cortex is, discussed in the context of the model's implementation.

What is Piriform Cortex of Pyriform Cortex

The piriform cortex, or pyriform cortex, is a region in the brain, part of the rhinencephalon situated in the cerebrum. The function of the piriform cortex relates to the sense of smell.

In human anatomy, the piriform cortex has been, described as consisting of the cortical amygdala, uncus, and anterior parahippocampal gyrus. More specifically, the human piriform cortex is located between the insula and the temporal lobe, anteriorly and laterally of the amygdala.

The function of the piriform cortex relates to olfaction, which is the perception of smell. This has been, particularly shown in humans for the posterior piriform cortex. The piriform cortex is among three areas that emerge in the telencephalon of amphibians, situated caudally to a dorsal area, which is caudal to a hippocampal area.

Farther along the phylogenic timeline, the telencephalic bulb of reptiles as viewed in a cross section of the transverse plane extends with the archipallial hippocampus folding toward the midline and down as the dorsal area begins to form a recognizable cortex.

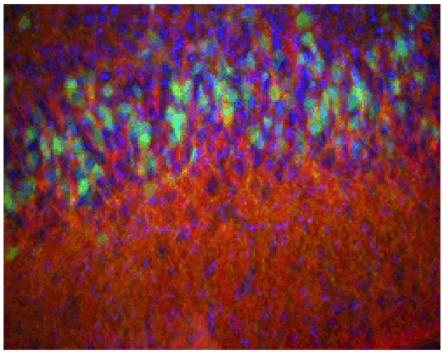


Figure 8-1. Piriform cortex from a 14-day-old D2-eGFP (green) mouse stained for enkephalin (red) and DAPI (blue) to show nuclei. Epifluorescence.

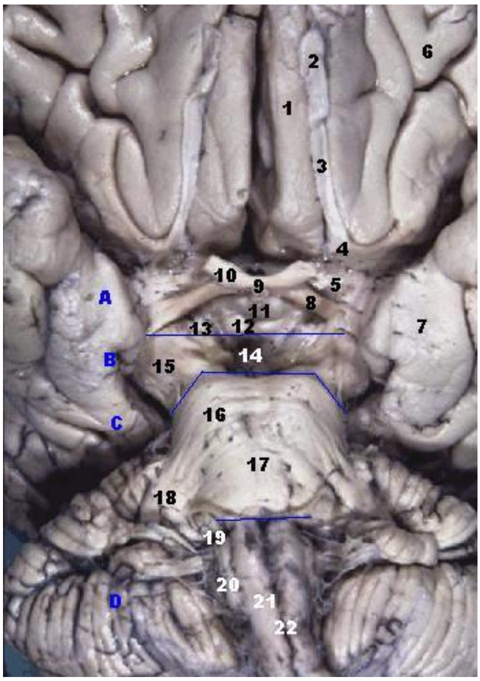


Figure 8-2. Human Brainstem Anterior (piriform cortex not labeled, but most of it is visible near #7).

8.3. STRUCTURE OF THE MODEL

Matt Wilson originally constructed the simulation discussed in this chapter when he was a graduate student at the California Institute of Technology. In fact, this simulation served as the initial basis for the construction of GENESIS itself. Portions of the model description below were, taken from a paper originally written by Wilson and Bower (Wilson and Bower 1989) [7]. The graphical interface was, later added to make the simulation user-friendly. Although this model has been, used to explore a wide range of cortical behavior (Wilson and

Bower 1989 [7], Wilson 1990 [8]), including associative memory function (Wilson and Bower 1988, Hasselmo, Wilson, Anderson and Bower 1990), the tutorial version has been, simplified for the sake of computational speed and pedagogical ease. In its current manifestation, the model allows you to reproduce experimental EEG patterns and to explore their possible physiological basis. Using later chapters of this book, the user can expand this simulation as he or she wishes.

8.3.1. Cellular Complexity

Pyramidal cells are the principal cell type in piriform cortex, and are, believed to be exclusively excitatory (Haberly and Price 1978, Haberly and Bower 1984). Superficial and deep pyramidal cells form, two distinct populations that differ significantly in their physiology (Tseng and Haberly 1989) [9]. There are also several populations of nonpyramidal cells or interneurons that can be, distinguished on anatomical grounds (Haberly 1983). These neurons appear to be GABAergic and seem to mediate both feedback and feed forward inhibitory effects.

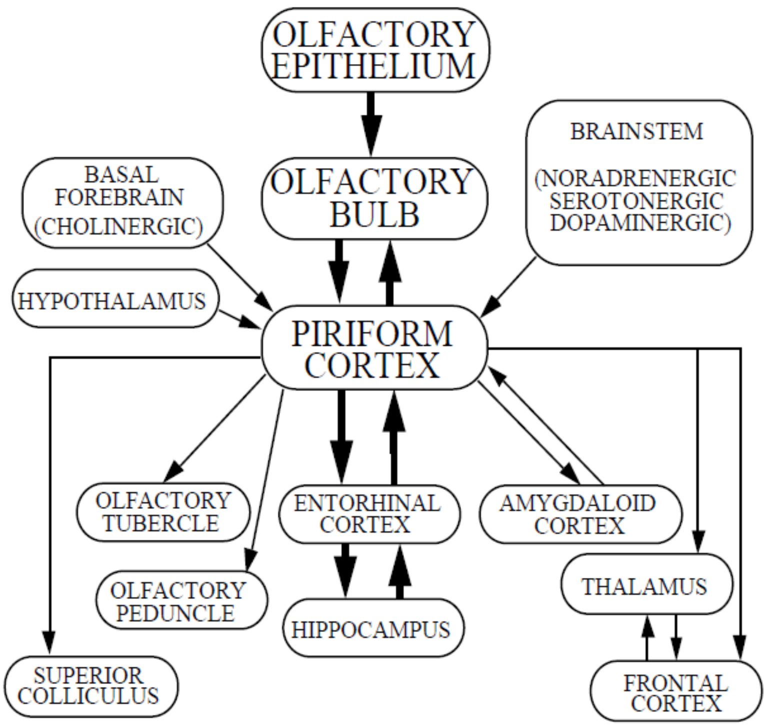


Figure 8-3. Piriform Cortex and Connected Areas.

As mentioned previously, computational limitations require that realistic network models be, constructed of fewer neurons than are actually found in the brain. The current model is, based on a single population of pyramidal cells plus two populations of inhibitory interneurons. A total of 135 neurons of each type are simulated, yet these neurons are intended to represent the full extent of the actual cortex (approximately 10 mm x 6 mm).

Accordingly, although we simulate neurons individually, the output of each neuron is, taken to represent the average activity of a larger group of cells that would normally be in its region. This adjustment for scale is, made in the strength of synaptic connections between, the cells and in the number of cells contacted by a particular neuron, within the simulated network.

Network models also usually include much simpler representations of neurons than are found in realistic single cell simulations. In the current case, pyramidal cells are modeled using five electrical compartments, whereas interneurons are modeled using only one. We have chosen to model pyramidal cells with five compartments for several reasons. First, in order to accurately, model field potentials, it is necessary to distribute synaptic inputs spatially along the dendritic processes of these cells. Secondly, the spatiotemporal distribution of synaptic input along the dendritic tree of the cell may be computationally relevant. Much more, realistic multi-compartment models are currently being, used to explore this possibility (Protopapas and Bower 1994) [10]. Little is lost in modeling inhibitory neurons as single compartments because little is known about the organization of their synaptic input and their small size and radially spanning dendrites preclude them from making significant contributions to the EEG.

In addition to the reduction in the number of compartments for each cell, the membrane properties of these neurons have also been, simplified. For example, although experimental evidence indicates that there are a number of Ca^{2+} and K^{+} currents in piriform pyramidal cells (Constanti and Sim 1987 [11], Constanti, Galvan, Franz and Sim 1985 [12], Constanti and Galvan 1983 [13]) in addition to standard Hodgkin-Huxley Na and K currents, none of these are modeled in any of the cells. Rather, a simple threshold criterion is, applied to the membrane potential to generate discrete spike events. The occurrence of spikes is, indicated with a spike waveform “pasted” onto the actual membrane potential at the appropriate time. In this way, the computationally expensive details of spike generation are, avoided but we are still able to generate the appropriate currents and membrane potentials associated with real action potentials. Similarly, synaptic conductances are modeled neglecting computationally expensive details such as the kinetics of ligand binding, neurotransmitter uptake, etc. Instead, changes in synaptic conductance are, modeled as the difference of exponential functions that approximates the shape of EPSPs seen in experimental studies.

8.3.2. Network Circuitry

Primary afferent input enters piriform cortex via the Lateral Olfactory Tract (LOT) projection from the mitral and tufted cells of the olfactory bulb. This is shown in Figure 9-4. Experimental evidence suggests that this projection is exclusively excitatory (Biedenbach and Stevens 1969a [14], Biedenbach and Stevens 1969b [15], Haberly 1973b [16], Haberly and Bower 1984 [17]) and extremely diffuse or non-topographic. These afferents make excitatory connections to pyramidal cells and feed-forward inhibitory cells in layer Ia (Haberly 1985) [5]. LOT input is modeled as a set of independent fibers that make sparse connections with pyramidal cells and both types of inhibitory interneurons. In this tutorial, activity along the afferent pathway is random over time. In both the actual cortex and the model, conduction velocities along axons are finite and vary with the axonal type (Haberly 1978) [18]. Signals travel along the LOT rostrally to caudally, and are distributed across the cortex via many small collaterals (Devor 1976) [19]. In the model, as in the brain, signals proceed along the

LOT towards the cortex at a speed of 7.0 m/s. Collaterals leave the main fiber tract at a 45^0 angle and travel across the cortex at a speed of 1.6 m/s (Haberly 1973a) [20]. In the biological cortex there is a diminution of afferent input to pyramidal cells moving rostrally to caudally that is reflected anatomically in the number of synaptic terminals (Price 1973, Schwob and Price 1978), and physiologically in the amplitude of shock-evoked potentials mediated by the afferent system (Haberly 1973a) [20]. To simulate this effect in the model, the strength of synaptic input due to afferent signals is exponentially, attenuated with increased distance from the rostral site of stimulation.

In addition to excitatory input from the bulb, pyramidal cells within the piriform cortex make excitatory connections with other pyramidal cells across the entire cortex (Biedenbach and Stevens 1969a [14], Haberly and Bower 1984 [16]). The fibers appear to spread out radially from the originating cell and travel rostrally at a speed of 1.0 m/s, and caudally at a speed of 0.5 m/s (Haberly 1973a [20], Haberly 1978 [18]) making local connections on basal dendrites of other pyramidal cells and distant connections on apical dendrites. In the model, fibers originating from pyramidal cells follow the same pattern of interconnectivity and signals are, propagated along each fiber with the corresponding delays. Simulation scaling considerations as described earlier require that association fiber interconnectivity be greatly, increased as compared with that of the actual cortex. As with different input, intrinsic excitatory connections, are, attenuated exponentially with distance from the originating cell.

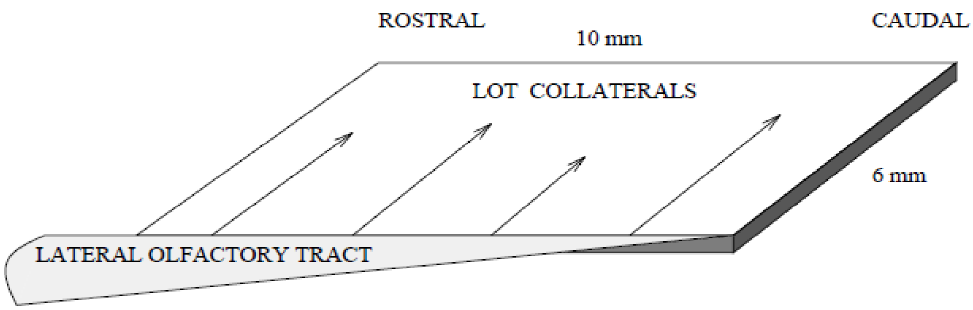


Figure 8-4. The Diagram Indicates the Distribution of Afferent Input to Real and Simulated Piriform Cortex.

Note that in Figure 9-4, Input from the olfactory bulb arrives via the lateral olfactory tract, which then sends perpendicular collaterals into the cortex that make sparse connections with piriform cells. The number of connections between the LOT and the cortex decreases as one travels rostrally too caudally.

Experimental evidence suggests the existence of two types of inhibition in the piriform cortex, both of which are, incorporated into the model. A well-documented Cl^- mediated feedback inhibition is, thought to be, generated by local interneurons that receive input primarily from local pyramidal cells as well as some afferent fibers (Biedenbach and Stevens 1969a [14], Biedenbach and Stevens 1969b [15], Haberly 1973b [21], Satou, Mori, Tazawa and Takagi 1982 [22], Haberly and Bower 1984 [16]). The outputs of these inhibitory interneurons feed back into nearby pyramidal cells where they activate a significant Cl^- conductance increase at the cell body. A K^+ mediated inhibition is also present in the biological cortex and appears to be generated by local inhibitory interneurons receiving primarily direct afferent input from the LOT as well as some association pathway input from

pyramidal cells (Satou et al. 1982 [22], Tseng and Haberly 1986 [23]). The outputs of these interneurons generate a long-latency, long-duration hyperpolarizing inhibitory potential in nearby pyramidal cells in the most distal part of the pyramidal cell apical dendrite. In the model, the K^+ mediated inhibition is, activated on the apical pyramidal cell dendrite by inhibitory neurons with both feed-forward and feedback input. The network circuitry for the model is, illustrated in Figure 9-5.

8.4. ELECTROENCEPHALOGRAPHY

Because the Electroencephalogram (EEG) is the physiological measure that this tutorial attempts to simulate, it merits some explanation. Unfortunately, the EEG is something which has a long history in neurophysiology but whose origins are still, debated. In general, physiological measurement techniques can be, grouped into those that record the responses of individual neurons, and those that measure the more complex aggregate electrical activity of networks of cells. Single cell electrical recordings can be either intracellular or extracellular and reflect the actual output of single neurons. The origins and significance of aggregate recordings such as the EEG are more difficult to determine. The EEG is identical to extracellular single unit recording in that it measures the field potentials generated in the space around neurons. It differs because EEG recordings represent electrical activity over a wider area of the brain. Typically, EEGs are, recorded from an array of electrodes placed on the surface of the brain or even the scalp. In this sense, one may think of the EEG as the average electrical activity of many neurons over a sizable area. The EEG is, calculated in the model using an array of 40 evenly spaced electrodes on the surface of the simulated cortex. Recordings from the array are, averaged to produce the EEG.

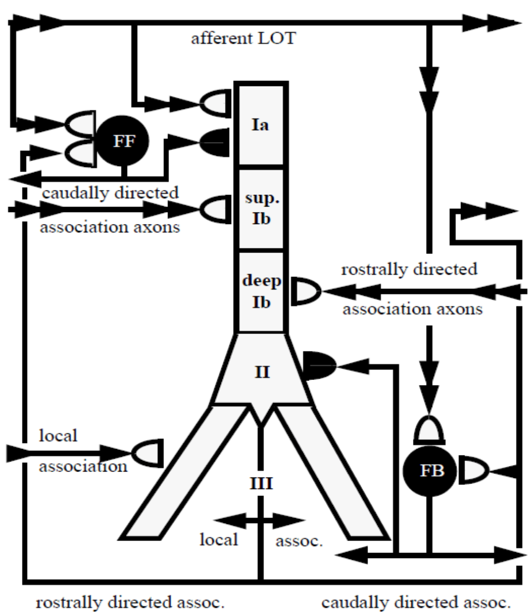


Figure 8-5. Diagram of Local Model Circuitry.

In Figure 9-5, the large cell in the center represents a pyramidal neuron. Inhibitory cells are, represented by black filled circles. The two types of inhibitory interneurons are distinguished as Feed-Forward (FF) and Feed-Back (FB). Single arrowheads signify local pathways. Double arrowheads are, used to show distant pathways. Excitatory synapses are, shown as lightly colored cones and inhibitory synapses are black. In the model, the basal dendrites are, modeled as a single compartment.

Understanding in detail how extracellular field potentials such as the EEG are, related to the activity of collections of single cells is not a straightforward matter. A neuron can be, thought of as a very complicated circuit consisting of resistors, capacitors, and batteries. Like any circuit, it must obey Kirchhoff's current law, which states that the sum of the total current entering and leaving a circuit node must equal zero. In the case of a neuron, this means that if synaptic current, for example, enters the cell at one point, it must leak from another, thereby generating an extracellular current (See Figure 9-6). An area of the neuron where current is entering the cell is, called a current sink. An area, where current flows outward is, called a current source. Here, we use the “physiologists’ convention” which holds that inward current is negative and outward current is positive. When treating current sources and sinks discretely as we do in the model, the field potential is dependent on these extracellular currents according to the equation:

$$\Phi(\vec{r},t) = \frac{1}{4\pi\sigma}\sum_{i=1}^{n}\frac{I_i(t)}{R_i} \qquad \text{Eq. 8-1}$$

where $\Phi(\vec{r},t)$ is the field potential in volts, $I_i(t)$ is the total current (amperes) from the i th current source into the brain tissue of conductivity σ ($\Omega^{-1}m^{-1}$), and R_i (meters) is the distance of the i th current source from the field point $\vec{r}$ (Nunez 1981) [24]. Although this equation is used to calculate field potentials in the model, it is based on the assumption that the brain is a homogeneous conductor, which is only an approximation to biological reality. The important thing to note here is that the size of the field potential increases in amplitude with the magnitude of extracellular current and decreases with distance between current source (or sink) and electrode. This has certain important implications. For example, it suggests that the action potentials of individual neurons often make little contribution to the EEG. Because the extracellular currents produced during spike generation are generally small, the greater magnitude of synaptic currents makes contributions that are more significant. Since the EEG represents the averaged electrical activity of many neurons, the more synchronous the activity, the stronger the signal will be.

Although understanding the basis for EEG activity is by no means straightforward, many attempts have been, made to correlate EEG patterns with certain types of animal and human behavior. EEGs are primarily, distinguished because of their frequency. The two frequencies most commonly discussed in the context of the olfactory system are the theta (4–7 Hz) and gamma (30–80 Hz). These types of EEG activity are common, not only in the olfactory system, but throughout the brain. They are also, found across species (Ketchum and Haberly 1991) [25]. In the rat, hippocampal EEGs in the theta range are prominent during exploratory behavior (Ranck 1973) [26]. More recent experiments have indicated that theta patterned

stimulation in the hippocampus is optimal for the induction of hippocampal long-term potentiation (Larson, Wong and Lynch 1986 [27], Staubli and Lynch 1987 [28]). Theta stimulation has been, used to successfully, induce LTP in the piriform cortex as well (Kanter and Haberly 1990) [29]. Interestingly, the theta rhythm approximates the rate of exploratory sniffing in the rat (Macrides, Eichenbaum and Forbes 1982) [30].

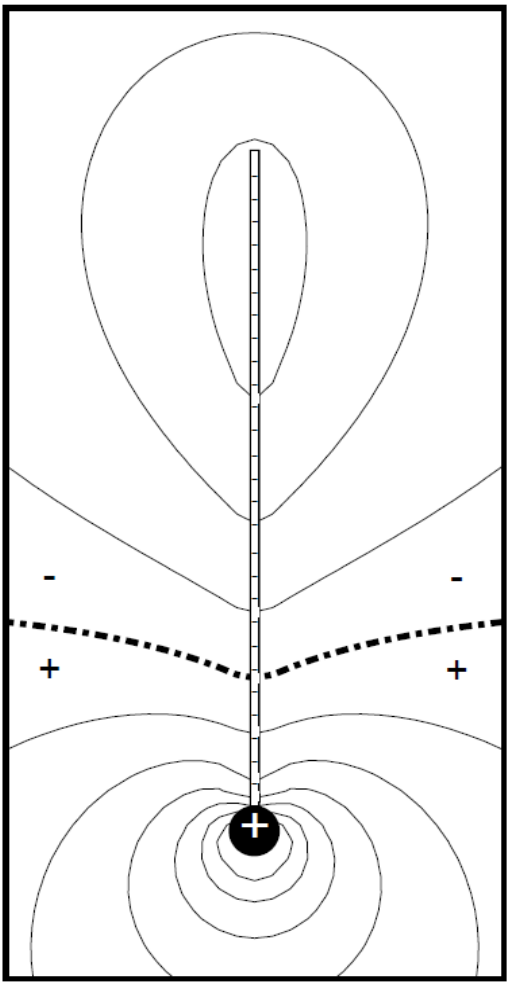


Figure 8-6. A model cell shows the distribution of current sources and sinks and extracellular isopotential contours at the moment, when the cell is receiving excitatory input all along its apical dendrite.

The gamma frequency (in the 40 *Hz* frequency range) has recently been the focus of intense experimental and modeling efforts. One of the earliest observations of the gamma frequency was in the olfactory system of the hedgehog in response to an odor stimulus (Adrian 1942) [31]. Since then, these oscillations have been, found in a variety of cortical areas. Numerous theories have evolved to explain the significance of the gamma rhythm. Researchers have proposed that it is a solution to the binding problem (Gray, Konig, Engel and Singer 1989) [32], a cortical information carrier (Bressler 1990) [33], and even a hallmark of consciousness (Crick and Koch 1990) [34]. Although such statements are highly

speculative, the ubiquity of these events suggests that the activity underlying the 40–60 *Hz* oscillations is, related in some way to neural computation. The research model on which this tutorial is based was used to explore the possible physiological underpinnings of these oscillations (Wilson and Bower 1991) [35] and led to the conclusion that cortical oscillations do not represent an information code, as suggested in some of the speculations above. Rather reflect the coordination of interneuronal communication within these networks (Bower 1995) [36]. A comparison of the EEGs generated by the model and real data is, shown in Figure 9-7.

In Figure 9-6, the thick contour shows the area of zero potential. Contour lines above the zero line give negative potentials. Below the zero line are positive potentials. Minus signs represent current sinks (current flowing into the cell) and the plus sign depicts a current source (current flowing out of the cell).

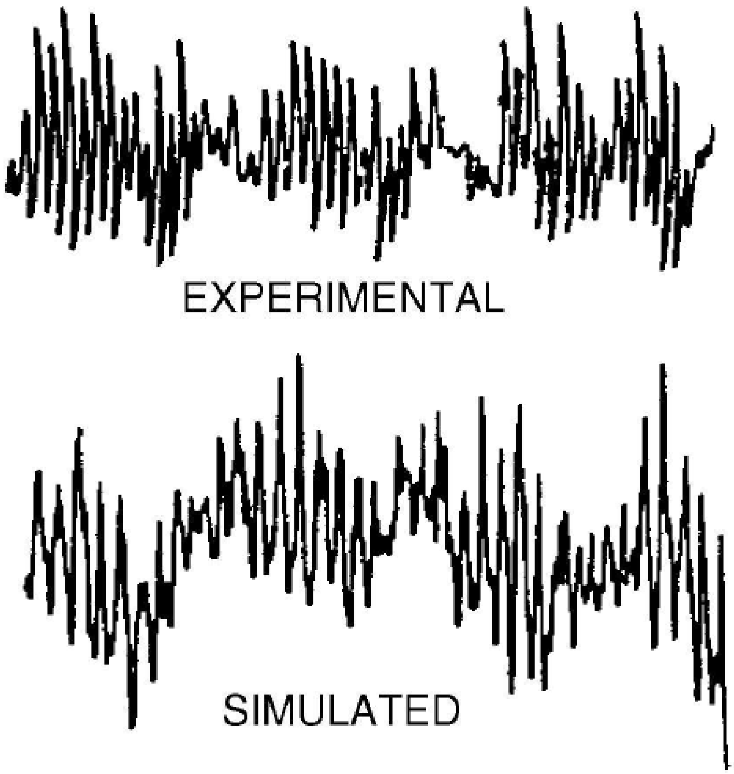


Figure 8-7. Comparison of an EEG measured from rat piriform cortex and data generated using this simulation (Wilson 1990).

8.5. USING THE TUTORIAL

As should be clear from the introduction to this chapter, the piriform cortex model that is at the base of this tutorial is quite complex. In fact, this is the most complex tutorial in this book. We have designed an interface for the tutorial that is relatively intuitive to use, however, as you will see, one can quickly become inundated with graphical displays and flashing colors. Any effort to understand all the details of this simulation is certain to be a

substantial undertaking. Accordingly, the rest of this chapter should be, regarded only as an introduction to some of the features of the tutorial. We encourage you to make further explorations on your own.

8.6. SUMMARY

The piriform model has been used to study a wide range of physiological and functional phenomena including associative memory function (Wilson and Bower 1988, Wilson and Bower 1992). The model attempts to retain biological plausibility, but simplifies the structure of the piriform cortex considerably. Despite the existence of a number of cell classes in the piriform cortex, only three were, modeled: superficial pyramidal cells, feed-forward and feedback inhibitory cells. Pyramidal neurons are, represented with five electrical compartments and inhibitory cells with only one. Although the piriform area contains approximately millions of cells, the model (in its tutorial manifestation) uses only 405. Axon conduction velocities and anatomical circuitry were fitted to experimental data in order to place realistic constraints on the network. Despite its simplicity, the model has been able to replicate a wide range of cortical behavior and to make experimentally testable predictions.

REFERENCES

[1] Haberly, L. B. (1990). Olfactory cortex, in G. M. Shepherd (ed.), The Synaptic Organization of the Brain, Oxford University Press, New York, chapter 10, pp. 317–345.

[2] Wilson, M. and Bower, J. M. (1992) “Cortical oscillations and temporal interactions in a computer simulation of piriform cortex,” *J. Neurophysiol,* 67: 981–995.

[3] Hasselmo, M. E. and Bower, J. M. (1993). Acetylcholine and memory, *Trends Neurosci.* 16: 218–222.

[4] Bower, J. M. (1995). Reverse engineering the nervous system: An in vivo, in vitro, and in computer approach to understanding the mammalian olfactory system, in S. F. Zornetzer, J. L. Davis and C. Lau (eds), An Introduction to Neural and Electronic Networks, second eds, Academic Press, New York, NY, pp. 3–28.

[5] Haberly, L. B. (1985). Neuronal circuitry in olfactory cortex: anatomy and functional applications, *Chemical Senses* 10: 219–238.

[6] Haberly, L. B. and Bower, J. M. (1989). Olfactory cortex – model circuit for study of associative memory, Trends Neurosci. 12: 258–264.

[7] Wilson, M. A. and Bower, J. M. (1989). The simulation of large scale neural networks, in C. Koch and I. Segev (editors), Methods in Neuronal Modeling, *MIT Press,* Cambridge, MA, chapter 9, pp. 291–333.

[8] Wilson, M. A. (1990). CIT Thesis, PhD thesis, California Institute of Technology, Pasadena.

[9] Tseng, G. and Haberly, L. B. (1986). A Synaptically mediated K^+ potential in olfactory cortex: characterization and evidence for interneuronal origin, *Soc. Neurosci. Abst.* 12: 667.

[10] Protopapas, A. and Bower, J. M. (1994). Sensitivity in the response of piriform pyramidal cells to fluctuations in synaptic timing, in F. H. Eeckman (ed.), Computation in Neurons and Neural Systems, Kluwer Academic Publishers, Norwell, MA, pp. 185–190.

[11] Constanti, A. and Sim, J. A. (1987). Calcium-dependent potassium conductance in guinea pig olfactory cortex neurons in vitro, *J. Physiol.* (London) 387: 173–194.

[12] Constanti, A., Galvan, M., Franz, P. and Sim, J. A. (1985). Calcium-dependent inward currents in voltage clamped guinea-pig olfactory cortex neurons, *Pfvlgers Arch.* 404: 259–265.

[13] Constanti, A. and Galvan, M. (1983), Fast-inward rectifying current accounts for anomalous rectification in olfactory cortex neurons, *J. Physiol.* (London) 385: 153–178.

[14] Biedenbach, M. A. and Stevens, C. F. (1969a). Electrical activity in cat olfactory cortex produced by synchronous orthodromic volleys, *J. Neurophysiol*. 32: 193–203.

[15] Biedenbach, M. A. and Stevens, C. F. (1969b). Electrical activity in cat olfactory cortex as revealed by intracellular recording, *J. Neurophysiol.* 32: 204–214.

[16] Haberly, L. B. and Bower, J. M. (1984). Analysis of association fiber pathway in piriform cortex with intracellular recording and staining techniques, *J. Neurophysiol.* 51: 90–112.

[17] Haberly, L. B. and Bower, J. M. (1984). Analysis of association fiber pathway in piriform cortex with intracellular recording and staining techniques, *J. Neurophysiol.* 51: 90–112.

[18] Haberly, L. B. (1978), Application of collision testing to investigate properties of association axons originating from single cells in the piriform cortex of the rat, *Soc. Neurosci. Abst*. 4: 75.

[19] Devor, M. (1976). Fiber trajectories of olfactory bulb afferents in hamster, *J. Comp. Neurol.* 166: 31–48.

[20] Haberly, L. B. (1973a). Summed potentials evoked in opossum prepyriform cortex, *J. Neurophysiol*. 36: 775–788.

[21] Haberly, L. B. (1973b). Unitary analysis of opossum prepyriform cortex, *J. Neurophysiol.* 36: 762–774.

[22] Satou, M., Mori, K., Tazawa, Y. and Takagi, S. F. (1982). Long lasting disinhibition in pyriform cortex of the rabbit, *J. Neurophysiol.* 48: 1157–1163.

[23] Tseng, G. and Haberly, L. B. (1986). A Synaptically mediated K^+ potential in olfactory cortex: characterization and evidence for interneuronal origin, *Soc. Neurosci. Abst* 12: 667.

[24] Nunez, P. L. (1981). Electric Fields of the Brain: The Neurophysics of EEG, Oxford University Press, Oxford.

[25] Ketchum, K. L. and Haberly, L. B. (1991). Fast oscillations and dispersive propagation in olfactory cortex and other cortical areas: A functional hypothesis, in J. Davis and H. Eichenbaum (eds), Olfaction: A Model System for Computational Neuroscience, MIT Press, Cambridge, MA, chapter 3, pp. 69–100.

[26] Ranck, J. B. (1973). Studies on single neurons in dorsal hippocampal formation and septum in unrestrained rats. I. behavioral correlates and firing repertoires, *Exp. Neurol.* 41: 462–531.

[27] Larson, J., Wong, D. and Lynch, G. (1986). Patterned stimulation at the theta frequency is optimal for the induction of hippocampal long-term potentiation, *Brain Res.* 368: 347–350.

[28] Staubli, U. and Lynch, G. (1987). Stable hippocampal long-term potentiation elicited by ‘theta’ pattern stimulation, *Brain Res.* 435: 227–234.

[29] Kanter, E. D. and Haberly, L. B. (1990). NMDA-dependent induction of long-term potentiation in afferent and association fiber systems of piriform cortex in vitro, *Brain Res.* 525: 175–179.

[30] Macrides, F., Eichenbaum, H. B. and Forbes, W. B. (1982). Temporal relationship between sniffing and the limbic theta-rhythm during odor discrimination reversal-learning, *J. Neurosci.* 2: 1705–1717.

[31] Adrian, E. D. (1942). Olfactory reactions in the brain of the hedgehog, *J. Physiol. (London)* 100: 459–473.

[32] Gray, C. M., Konig, P., Engel, A. K. and Singer, W. (1989). Oscillatory responses in cat visual-cortex exhibit inter-columnar synchronization which reflects global stimulus properties, *Nature* 338: 334–337.

[33] Bressler, S. L. (1990). The gamma wave: a cortical information carrier?, *Trends Neurosci.*13: 161–162.

[34] Crick, F. and Koch, C. (1990). Towards a neurobiological theory of consciousness, *Sem. Neurosci.* 2: 263–275.

[35] Wilson, M. and Bower, J. M. (1991). A computer simulation of oscillatory behavior inprimary visual cortex, *Neural Computation* 3: 498–509.

[36] Bower, J. M. (1995). Reverse engineering the nervous system: An *in vivo*, *in vitro*, and *in computo* approach to understanding the mammalian olfactory system, *in* S. F. Zornetzer, J. L. Davis and C. Lau (eds), *An Introduction to Neural and Electronic Networks*, second edn, Academic Press, New York, NY, pp. 3–28.

[37] Wilson, M. A. (1990). *CIT Thesis*, PhD thesis, California Institute of Technology, Pasadena.

Chapter 9

NEURAL NETWORKS AND FUZZY LOGIC SYSTEMS

Most scientists and researchers in field of neuroscience, machine learning and smart generation of new computers are already applying neural networks in series with fuzzy logic system, which considers the use of fuzzy inputs and outputs for neural networks to the employment of individual neural networks to quantify the shape of a fuzzy membership function. However, integration of the field of neural networking and fuzzy systems, into something as new technology known as "neurofuzzy" holds even greater potential benefits in reducing computing time and make this close to real time as much as possible is there and results in better optimization of data processing and decision making. Such process is much, needed to build a resilience system that acts in real time for stakeholder to take a proper counter measure against harming events or adversary effects measures to their business or for that matter make an on time decision to improve day-to-day operation for better results.

9.1. INTRODUCTION

In world of fuzzy thinking, in parallel to the development of neural network theory, along with *fuzzy theory or fuzzy logic* (FL) utilizing both first and second types fuzzy logic.

As we have mentioned in Chapter 6, the primary and essential information processing structures of interest in neurocomputing are *neural networks*, although other classes of adaptive information processing structures are sometime need to be, considered as well. This includes machine learning such as automation, genetic learning systems, data-adaptive content addressable memories, simulated annealing systems, associative memories, and fuzzy learning systems, based on fuzzy logics both first and second types.

Fuzzy logic along with neural networking driving the future artificial intelligence is tremendous improvement for any autonomous system acting as point of real time decision making for, any today's fast paced organization to deal with friendly type events or adversary one as well, for their given applications. These organization could anyone in financial or banking market, homeland security, intelligent communities, defense, etc that deal with the adverse measures. However, in case of supply chain perspective looking natural disaster might be as important as manmade one, where these natural disasters have negative impacts on their inventory control and supply chain at point of sale or manufacturing within their final assembly line for their final productions.

In this chapter, we try to have a better understanding of Fuzzy Theory and why is the term *fuzzy* is used, although we had a good description fuzzy logic concept in Chapter 2, we just try to briefly remind the reader on such summary. "Fuzziness" can be, found in our day-to-day decision makings, in our thinking, in the way us as human process information, and particularly, in our language; statements can be unclear or subject to different interpretation.

To have a better concept for the fuzziness, we can think of phrases like "see you later" or a little more or less," or I do not know that much" are typical fuzzy expression that are falling in fuzziness categories. This is because the ambiguities that are involved in any of these phrases such as "later," "more or less," "that much" and are stems from the different interpretations or perceptions. For example, if we look at the statement "see you later" form an engineer point of word of "later" may be anywhere between nanoseconds to few seconds time interval, but for someone else in different such as paleontologist, it may be on the order of thousands of years. As you can see, the order of magnitude is relative, therefore, if *some* fuzzy units are used, one should look at it within its context and find a point of reference and a measuring scale or units.

From time to time, the fuzzy statements can be made in some kind of comparisons form, which we may call it relative units and sub-units that are not indication of absolute units. To clarify this, we can use the following examples. Consider "Runner **A** is fast," "Runner **B** is faster than **A**", and Runner **C** is slower than **B.**" In this example, we make, two observations and that are: Fuzzy statement may establish *taxonomy* that (**B** is faster than **A**, and **C** is slower than **B**) or *amb*iguity, (which is not clear if **A** is faster than **C**) and there is not enough information to measure the speed of **A**, **B**, or **C**.

In another example, consider the statement that if we say, "Mark is very tall" is a fuzzy statement, simply because we have no frame of reference to measure his tallness, how tall he can be in compare to whom. For example in almost any basketball team the average height of players are 6ft 2 in, therefore in that perspective "very tall," most likely means taller than 6 ft 2 in. To average person, "very tall" often means taller than 5 ft 8 in, often but not necessarily 6 ft 2 in.

Fuzziness is often confused with probability, which means a statement probabilistic if it expressed a likelihood or degree of certainly or if it is the outcome of clearly defined but randomly occurring events [1].

For example, making a statement such as "There is 50/50 chance that I will be there" is purely probabilistic. Probability itself can have some degree of fuzziness. In the statement "Most likely I will be there," all odds have been mentally weighed and some degree of certainly or probability has been expressed. On the other hand, the statement "I may be there" expresses complete uncertainty, undecidability, and, hence, fuzziness [1].

To further, clear the crispness of logic versus fuzzy logic, you are probably familiar with logic that has well-defined decision levels or thresholds that are, known as Binary and Multivalue. *Boolean* or *Binary Logic* is, based on two *crisp extremes*--YES or NO or 1 -- 0. This is where the difference between Boolean Logic (BL) and Fuzzy Logic come about. In Boolean Logic, the situation of interest either is *False* or *True* and on other hand in case of Fuzzy Logic (FL), things are either partially *True* or partially *False*. In this context, the YES or NO is an answer beyond any doubt. *Trivalent Logic* (TL) is logic of three definite answers, such as empty or half full, and full or in computer logic 0 or 0.5, and 1. The binary numbers 1 or 0, or 1, 0.5, 0 in trivalent logic represent normalized thresholds. Similarly, the multivalue logic has many well-defined threshold levels.

Fuzzy logic, however, has *unclear* thresholds. For example, if we take the trivalent logic and *fuzzify* it (i.e., change the crisp thresholds to obscure ones), the values of the thresholds can be stated as a range of values. The crispness of the numbers, 0, 0.5, and 1 may be replaced by “from 0 to about 0.4”, “from about 0.2 to about 0.8”, and from about 0.6 to 1”, respectively. This is also can be cleared with the following example, and that is if you look at three distinct dots through a well-focused camera lens, you will observe the dots with crisp perimeters. If the image is out of focus, however, the dots become unclear and “fuzzy,” and perhaps overlapping each other. This action is termed *fuzzification* and is fuzzy control systems is routinely, done to focus the lenses of camera on the three dots.

We can also introduce another example of fuzziness, utilizing string theory. For example, consider a wood log of 2x2 Lumber and when you look at it at close distance from your eyes, it seems like a three dimensional object. However, if you keep moving it away from you or you move away from it a two-dimensional object long distance seems like a two-dimensional object and finally if the distance between you and lumber becomes long enough then you would see it as a one-dimension object. If you ask someone, whose is standing next to you at that far distance from lumber, and ask, what is the dimension of the object in front of your eyes, you will most probably hear, a one-dimension object and you would know for sure that is not the case. Off course, if you bring lumber closer to that observer, eventually he or she would see as a two-dimensional object and finally, observer can see it as true three-dimensional object if the lumber comes closer to that observer.

Fuzzy logic has recently attracted a lot of attention and so many applications have been, developed around this interesting field of science. It has been, applied into military intelligence, homeland security, deep machine learning data processing in banking, the stock market and even when it comes to controlling the a machine as simple as dishwasher and laundry machine logic of fuzzy has been used to control cycling process for their operation from end to end. In communications process it has been, used on the systems level and in signal transmission. In Artificial Intelligence and remote sensing, fuzzy logic is now playing an important role and on the system level, fuzzy logic applications determine the best parameter values for call switching, call routine, system reconfiguration, and so forth and so on.

In remote sensing, pattern recognition, and signal processing as well. Fuzzy logic application, in case of communication and signal processing, determines the degree of the fuzzified received signal or distortions due to environmental variation, electrical interferences, medium mismatches, and other parameter such as permeability and permittivity of media where the wave signal passes through and then “defuzzify” the signal.

In nutshell, fuzzy logic is a powerful tool for the intelligent retrieval of non-statistical, ill-defined information in static, sequential, and real-time applications.

9.2. Fuzzy Logic, Driving Neural Networks System

As we have learned so far through previous chapters of this book, Artificial Neural Networks (ANN) and Fuzzy Logic (FL) work together, artificial networks classify and learn rules for fuzzy logic and fuzzy logic infers from unclear network parameters.

Historically, in any new development in science and technology, often comes from observation made from a different perspective. For instance, we gain information into the behavior of dynamic system by viewing it as being in the “time domain” and or the “frequency domain.” This could be clear to you as reader, by looking at the statisticians of recent decades, dealt with autocorrect ion and cross-correlation functions in the time domain. On the other hand, electrical engineers challenged with power and cross-spectral densities in the frequently domain without either group to realize that these two concepts are interrelated to each other and was driven by famous relation, proposed by French mathematician Fourier, which is know as Fourier Transformation.

These both group of researchers and scientists have found that analysis of the fluctuation in process variables provides useful sets of information about the variables as well as the processes that are involved. These fluctuations, which results in uncertainties in measured variables, often are caused by some sort of random driving function, and as example we can look at turbulences in fluids, signal processing in communications, rotational unbalance and unstableness, etc. [2].

Interoperability of fuzzy systems principles and neural networking are presenting two distinctive and yet very flexible methodology that provides a robust system and particular a more intelligent artificial and bio computing for futuristic autonomous system and consequently a more bulletproof resilience systems and they are able to deal with uncertainty as real time as possible [3].

The uncertainties that are, important include both those in the model or description of the systems that are involved as well as those in the variables. The uncertainties and unexpected events, in particular if they are adversary, and needs a counter-measure to be, taken to neutralize them before they become harmful, including their variables that are generating it as well.

In addition, these uncertainties usually arise from system complexity that are, nonlinear and inhomogeneous and beyond control variables are involved and they change near real time of not real time. We can think of complexity here, as a property of system description--that is, related to the means of computation or language for deep learning for example and not merely a system's complicated nature. Neural networks approach along with Fuzzy Logic System (FLS) in place drives the modeling that is representation by using precise inputs and outputs, which are, used to train a generic robot. However, if this robot has enough intelligent built into it and has sufficient degrees of freedom to be able to formulate a good approximation based on fuzzy system to come up with good approximation of this complex relationship between the inputs and outputs. This type of resilience system, then suggests, the best possible solution around it to stakeholders and decision makers of the business where these robots are, implemented as part of their applications, which were, mentioned in previous section of this chapter.

Bear in your mind, in fuzzy systems, the reverse situation prevails, which means the input and the output variables are, encoded in “fuzzy” representations, while their interrelationships take the form of well-defined Boolean logic of *if*/*then* rules. However, ingenious observation and formulating of fuzzy logic by Zadeh is that, the uncritical pursuit of precision may be not only, unnecessary but actually, a source of error led him to the formulation and notation of fuzzy logic sets.

In corporation of neural networks and fuzzy logics, allow, fuzziness in this interoperability and means more flexibility in the definition of system, that boundaries may be

described more generally, and crisply, however, in case of Business Resilience System (BRS) a threshold may be set via Service Level Agreement (SLA) [3]. Inputs may be, described more vaguely, yet better control may be, obtained.

The network itself may be fuzzy, not well defined, and able to reconfigure itself for best performance, simply because what we have described in Figure 6-18. As it can be, seen in that figure, for best machine learning wit deep learning capability, still it is fuzzy scene to recognize the power poll from trees.

9.3. THE FUTURE PATH FOR FUZZY NEURAL NETWORKS SYSTEMS

Everyone knows about the old science of Boolean and multiple value logic, which have used by many scientists for many decades, before Fuzzy Logic was, introduced by Professor Lotfi Zadeh of UC Berkeley. As we mentioned previously, Fuzzy Logic (FL) follows the same path as Boolean Logic (BL) by initially, binary logic started as a linguistic set of statements, such as if **A** = **B**, and if **B** = **C**, then we can say **A** = **C** is true fact.

However, the mathematical notation translated the linguistic statements into equations and theories were, developed that are being, taught by subject matter experts today. These theories have been, applied successfully in the development of many logical applications.

However, the fuzzy logic also started as a linguistic set of statements. For example, if **A** is taller than **B**, and **B** is shorter than **C**, what is **A** with respect to **C**?. As it can be, found in many mathematical articles and theories in this filed, we can come to conclusion that, we may make reasonable argument and extrapolation and deduce that fuzzy logic will prove itself as a binary logic as Boolean logic has done.

The neural networks and fuzzy logic complement each other to the maximum of the both worlds. A fuzzy concept that is, fused with process of human “thinking” is suggesting a better and superior technology for future of artificial intelligence and consequently smarter business resilience systems [3].

The above given statement, can be validated by various logic integrated into most semiconductor circuits chip, such as High Array Logic (HAL) and Programmable Array Logic (PAL) bipolar processing and other CMOS circuit manufacturing that are operating under fuzzy controllers type machinery such typical home made dishwasher or laundry machine. This also can be, extended to the applications of in auto-industry for the automobile engine control, or futuristic autonomous vehicles, robot control in AI and BRS, cameras for film and video, appliances, such as thermostat control heater or cooler, military, banking, homeland security, supply chain type operation and so on [1].

Additionally, numerous fuzzy algorithmic solutions have been, suggested by subject matter experts in the field of fuzzy logic beyond just the hardware applications and solutions, and they include, signal processing, pattern recognitions, machine learning, image process for remote sensing and others

Banking industries including Federal Reserve, are looking at Business Resilience System (BRS), by processing huge volume of incoming data at the level of Big Data size, as part of their day-to-day Customer Relation Management (CRM) and other operating applications that is driven by fuzzy logic [3].

As we have seen in this book so far, fuzzy logic and neural networks combined are one of most important characteristics of artificial neural nets as part of their inputs classifications. With this interoperability, it is useful if the plasticity is, maintained, which means then the ANN can continuously classify and update classifications as well.

However, fuzzy systems and neural networks each have their own shortcomings. When one designs with neural networks alone, the networks is a black box that needs to be, defined clearly per its application requirements. This is highly compute-intensive process. One must develop a good sense, after extensive experimentation and practice, of the complexity of the network and learning algorithm to be, used and the degree of acceptable for the application under consideration.

Going forward with design of fuzzy systems, in future both from tactically and strategically, we need to have a thorough understanding of fuzzy variables and membership functions, of input-output relationships as well as the good choice of fuzzy rules that contribute the most to the solution of the application.

As we have indicated in above and previous chapters of this book, neural nets and fuzzy system are very different with each other, but they have very close relationship and complement each other to maximum extends. They both can work with imprecision in a common space that is not, defined by crisp and deterministic approach of Boolean logic, however they have very similar common denominators from application perspectives.

The shortcomings of neural networks and fuzzy systems may be, overcome if we incorporate fuzzy logic operations into neural networks and learning as well as classification of neural networks into fuzzy systems. The result will take to different aspect of ANN, which is, called Fuzzy Artificial Neural Network (FANN) [1].

REFERENCES

[1] Stamatios V. Kartalopoulos, "Understanding Neural Networks and Fuzzy Logic, Basic Concepts and Applications," *IEE press*, 1996.

[2] Lefteri H. Tsoukalas and Robert E. Uhrig, "Fuzzy and Neural Approaches in Engineering," John Wiley and Sons, Inc. 1997.

[3] Bahman Zohuri and Masoud Moghaddam, Business Resilience System (BRS): Driven Through Boolean, Fuzzy Logics and Cloud Computation: Real and Near Real Time Analysis and Decision Making System 1st ed. 2017 Edition.

Chapter 10

THE EXTRAORDINARY FUTURE OF ARTIFICIAL INTELLIGENCE

We are living in the computer age now, but so far, that fact has seemed irrelevant to the question of life after death. Sure, computers may have some place in medical technology, but they will not enable us to live forever. Some computer scientists say this latter claim is false--that in the next century computers will enable us to live forever! We will not have to wait for our bodies to degenerate to the point where they bring about death. Prior to such a point, each of us will be able to transfer his or her mind to a robot and continue living in the robot body. This chapter is duplication of Dr Winfred Phillips thesis and part of the write up by Professor David Leech Anderson of Illinois State University, Philosophy Department and is, reprinted with permission.

10.1. INTRODUCTION

People have long speculated about the possibility of creating machines that are as intelligent as human beings are. Likewise, people have speculated about the "mechanical" nature of human beings. Are humans "nothing more than" organic machines? Could a steel and silicon machine ever acquire the properties necessary to qualify as a "person"?

Speculation about these questions is nothing new. What is new is the number of well-respected researchers in robotics, engineering, and artificial intelligence who have predicted dramatic breakthroughs in the next fifty years that will (or so we are told) answer many of these fundamental questions, once and for all. If these prognosticators are right, then human life -- and "machine life" -- on this planet will never be the same.

In the past decade, a number of prominent figures in areas of advanced technology have begun making two remarkable claims. The first claim is that in this century, possibly in as little as 50 years, there will be machines that are more intelligent (not to mention, more powerful) than humans. The second claim is that in that same period, the technology will be available to download the human "mind" and re-install it in a computer. The result is that when your biological body wears out, it will be possible to relocate your "mind" (the real you?) into a mechanical body with a computer brain. You will become *immortal*. Hardware can be, replaced when it wears out and back-up copies of your "mind" will be available

whenever the software is corrupted. Moreover, they say, this will not happen in some distant future, but for many of you, dear readers, within *your* own lifetime.

This is heady stuff. In addition, as you are probably already aware, the claims are being, made are not merely empirical claims about what level of performance machines may soon achieve, but they are also robust and very controversial philosophical claims about what it means to be a "person," and, more particularly, what it means to be the very person that is "you." It is an empirical question whether the next two generations will produce a robot whose behavior is so complex and so subtle that it will be, quite literally, indistinguishable from the behavior of normal human beings. It is a philosophical question whether having the property, "behaving in such a way as to be indistinguishable from normal human beings" is all that is necessary (i.e., a "sufficient condition") for having the moral status of being a person.

Likewise, it is an empirical question whether the technology will ever be available to "read" your memories and your personality right off of the neurons in your brain so as to preserve in computational form the information contained in your memories and the behavior-patterns reflective of your personality. However, it is a philosophical question whether the resulting robot that has information about your past experiences and that shares behavior-patterns with your prior self is indeed *you* or is rather a mechanical imposter which is, infuriatingly enough, acting as if it is you.

Before going any further, you might wonder whom the people are who are making such extraordinary claims and on what grounds they are making them. I will mention three of the most prominent people who have done much to set the course of the present debate. Hans Moravec has been a pioneer in robotics and AI at Carnegie Mellon University for many years. He is also a favorite interview subject for many of the television programs that have charted the progress of AI and robotics during the past couple of decades. As early as 1988, he published a book, *Mind Children: The Future of Robot and Human Intelligence* where he began to defend his bold claims. In 1999, he published another book, *Robot: Mere Machine to Transcendent Mind* where he continues the story. Joining Moravec, is Ray Kurweil, an accomplished inventor and engineer who is responsible for major breakthroughs in voice recognition, synthesized speech production, to name a couple. In 1999, he received the National Medal of Technology, the nation's highest honor in technology, from President Clinton. The publication of his book, *The Age of Spiritual Machines: When Computers Exceed Human Intelligence* (1999) received a good deal of fanfare in the mass media, spurred on by Kurzweil's own very impressive website which he uses to great effect to promote and advance his views.

Moravec and Kurzweil are both, well respected in their fields and, when they make bold pronouncements about the future of robotics, people listen. Yet, they are both, known for being, shall we say, "enthusiastic" about their views and so many readers have taken their startling predictions about the future with a grain of salt. It was a matter of some note, then, when Bill Joy, co-founder and senior research scientist at Sun Microsystems, advanced many of the same predictions. Joy is a man well respected in the industry and considered a sober analyst of the current state of technology, not prone to flights of fancy. It came as a bombshell, then, when Joy published an article in Wired magazine (the cover story), titled "Why the Future Doesn't Need Us" in which he concurs with many of the claims of Moravec and Kurzweil. Joy is not so optimistic about a future filled with intelligent robots. Rather than looking forward to a time when robots will bring with them a land of plenty, he worries that

they may bring our demise. Smarter and more powerful robots may care little for the well-being of humans. Joy goes so far as to suggest that computer scientists and engineers presently working on advanced AI and robotics projects are in a situation similar to the physicists working on the Manhattan project. They are, faced with a moral dilemma: Should they contribute to a venture that may lead to the production of machines that will destroy the human race?

(Copyright Nancy Stahl, Used with Permission).

Figure 10-1. Futuristic Robots.

However, is this really worth worrying about? Is there any reasonable chance that robots will in our lifetimes - or even in this century - become as intelligent as human beings become? Is it at all plausible that you might one day be a robot?

Many people, including some of the authors mentioned above, too often make it seem that the main obstacle that must be overcome is the production of a computer fast enough to perform the same number of operations per second as the human brain can perform (in the firing of its neurons). Surely, creating a fast computer will be the easiest task to accomplish. With all the progress that has been made in AI and robotics in the past 30 years, we are still a long way from creating a computer that can perform the kind of higher cognitive functions that humans perform. There are two dominant computer paradigms -- symbol-processing digital computers and connectionist networks

Neither is currently capable of producing anything like human-level cognitive performance. Traditional AI programs have provided us with "artificial agents" that can manipulate linguistic symbols and so, in a sense, "speak" a language. But, these programs can

only operate "intelligently" within a very narrow range, and when you try to *scale them up* to include anything genuinely complex, their performance suffers. As Alan Turing (he of the famous "Turing Test" for machine intelligence) has argued, narrow intelligence is no intelligence at all. The very nature of intelligence is to perform rationally within a broad domain.

None of this is to say that computers will not achieve human-level intelligence. We are still at the very beginnings of the computer age, and it is impossible to predict what new strides will be possible with:

1) Variations on the symbol-processing or the connectionist network models,
2) Hybrid systems that combine the strengths of both paradigms, or
3) Some third computer model, like, say, one based on dynamical systems theory.

However, the emphasis, here, is on the "impossible to predict." we do not believe that present achievements support the view that this kind of performance can be, expected at any particular time in the future.

For philosophers, though, there are many other questions that must be answered other than questions about the outward performance of computers. The claims about human immortality, and the prospects of downloading your mind into a computer, hang on far more than claims about outward performance (what we might call the machine's behavior). The issue, of course, is that a machine might perform *as if* it has beliefs, as if it has hopes and dreams, and *a*s if it experiences pleasure and pain, and yet it may not possess any of those properties. It may be simulating them without genuinely possessing them. To determine whether a thing has a genuine mental state, we must know the fundamental nature of mental states. That requires a theory - a theory of mind, as philosophers call it. If the theory called functionalism is true, then machines with all of the properties that we have should be a real possibility. Nevertheless, is functionalism true? It remains a matter of considerable controversy.

Can robots achieve human-level intelligence? Can robots possess the rich range of mental states that we do? Even if these questions are, answered in the affirmative, there are further considerations that must be, rose before we can have grounds for optimism about the possibility of downloading your mind into a robot and achieving immortality. If the contents of your mind if your memories, beliefs, and personality if the very person that is **you** is to be downloaded into a robot, we must have some plausible theory of "personal identity," some theory that tells us what are the necessary conditions for "being you." Just as philosophers, have been arguing about theories of mind for centuries, so they have been arguing about theories of personal identity.

One of the simplest theories of personal identity is the memory theory, defended by John Locke. This might be just the ticket for those hoping for robot-immortality. For a robot in the year 2050 to be the same person that I am today, it is only required that the robot have at least one of my memories. "That seems easy," we hear you say. All that is required is that one of my memories (for example, some fact about my past that is currently stored in my brain) be encoded and stored on the hard drive of the robot's computer. However, of course not all encoding, of facts on a computer count as genuine memories. We can write the sentence "we saw John at his 6^{th} birthday party" on the computer on which I am writing this essay. It is then a fact about my past stored on the hard drive of a computer. However, it hardly follows that

my laptop **remembers** seeing John at his 6th birthday party. What distinguishes, then, between the real memory of a particular person and mere stored information? That, of course, is precisely where debates about the memory theory of personal identity get interesting.

10.2. THE EXTRAORDINARY FUTURE

The average human life span is less than a hundred years. If we are lucky enough to make it that far, then our bodies are so, worn out that our earthly lives soon ending. Many of us succumb to some disease or other and do not make it that far.

Some religions or their followers make claims about life after death, but of course in the case of any particular religion or its claims there are other people who do not believe those claims are true. Actually, some people believe that all such religious claims about life after death are false or dubious. There do not seem to be any religious claims to immortality that are, accepted universally. Therefore, it is not, uncontroversially true that there is life after death.

We are living in the computer age now, but so far, that fact has seemed irrelevant to the question of life after death. Sure, computers may have some place in medical technology, but they will not enable us to live forever.

Some computer scientists say this latter claim is false--that in the next century computers will enable us to live forever! We will not have to wait for our bodies to degenerate to the point where they bring about death. Prior to such a point, each of us will be able to transfer his or her mind to a robot and continue living in the robot body.

In this chapter, the author Dr Winfred Phillips will examine the plausibility of the claim that mind transfer from a human to a computer (robot) will occur before the end of the twenty-first century. I seek to determine to what extent we can be confident that mind transfer will occur during the next century. To do so I will examine relevant technological issues as well as philosophical issues pertinent to this claim. If a statement is, needed of the "thesis" that I will defend, it is that the authors who depict the near arrival of human-computer mind transfer are too optimistic in their prediction.

However, he says at the start that my perspective will be both sympathetic and critical. Several authors have developed a general line of argument supporting the claim of the impending possibility of human-computer mind transfer. I will examine this line of argument critically to see if it really sounds plausible. But where it does not, I will sometimes try to explore options for shoring it up. I have no hidden political or religious agenda in support of or against human-computer mind transfer. All I want to know is to what extent we can have confidence that such a thing could happen in the next century. This thesis will argue that we should not be confident.

As Dr Phillips claims, he does not know whether you wish to consider the prospects for human-computer mind transfer--perhaps you think the claim is so preposterous that you have dismissed the possibility of even considering it. I do not think we can know that it will never happen. Perhaps you will be interested in at least is considering the question after reading the claims of those who think it will happen. Even if you do not care about human-computer mind transfer, you might nevertheless be interested in the technical and philosophical issues discussed in my investigation.

The claim is that a human being will be able to transfer his or her mind to some kind of smart or potentially smart robot and continue existing as the same person but now in the robot brain and body instead of the old human brain and body. If we step back from the claim we can see that a number of further issues and questions arise immediately. Is the human mind such that it could be transferred? What is the human mind anyway--is it the brain or something more? By what technical means could we get that mind out of a human brain and body so to transfer it? By what technical means could we put it in a computer or robot? What would the robot have to be like in order to "receive" that mind?

Moreover, after the purported transfer, would it really be the same person in the robot body, that was in the human body? In other words, would it really be the transfer of yourself, to yourself, rather than the creation of some new person?

With so many issues and questions, we could start in any number of places. I choose to proceed in somewhat the same order that most authors have adopted in arguing for the claim. They don't first get involved in a complicated discussion of the human mind and whether and how it could be captured and transferred. They do not start by discussing the issue of personal identity. Rather they start by trying to convince you that robots in the twentieth century will get really smart, as smart or smarter than humans.

Winfred Phillips can think of two reasons they start here. First, they want to get you excited about robots. If they can convince you that robots in the twenty-first century will be really smart, smarter even than humans, then you will be more enthusiastic about the technological possibilities and perhaps see claims about human-computer mind transfer as not so far-fetched after all.

A second reason for starting here is that the authors who argue for the impending reality of human-computer mind transfer often have other matters on their agendas. Not everyone who writes about it is discussing just this subject. These writers vary in the extent to which human-computer mind transfer is seen as a major theme in their depictions of life in future centuries. Their books are not solely about human-computer mind transfer but also about how advanced computers and robots will soon be. They think robots will be so, advanced that some will dwell in society much like humans--walking and talking, living their own lives, etc. In future centuries, robots will even colonize space. Discussions about human-computer mind transfer often occur in the midst of this kind of advancement of a larger robot program, so the natural place to begin their discussions is with the robots themselves.

How does human-computer mind transfer fit into the robot program? As it turns out, humans might miss most of the "action" if they remain limited to their current bodies and capabilities. We are not nearly as smart as smart beings could be, and our bodies are not all, that capable or durable. If we do not want to miss most of the action, we really need to find a way to overcome the limitations of our human bodies--we need to get smarter, stronger, and more durable. Human-computer mind transfer is, seen as the way to do that.

He says, I am not considering the larger question of the future of robots in general, nor am I trying to get you excited about robots, so I do not have to discuss all the claims and arguments about robots that occur in the writings of the authors we will examine. Neither do I have to follow their order of exposition, but in explaining their position on human-computer mind transfer it will turn out to be convenient to start with robots. Many of the technical issues that come out in the robot discussion will be useful later. So first, Phillips will consider the question of where robots will be in the next century. Then I will examine the issue of what the mind is and whether, it is the sort of thing, a robot could have. I will next consider

possible mechanisms by which a transfer could take place. Finally, I will discuss the issue of personal identity--will it be the same person after the transfer?

He thinks that for a human person to transfer his or her mind to a computer and continue personal existence in a robot body, the following predictions must be correct:

1) Robots will be as smart as humans are.
2) Robots will be capable of being persons.
3) There will be a viable mechanism for the transfer of a human's mind from a human body to a robot body.
4) Transferring one's mind to a robot will allow one to continue one's existence as the robot.

For human-computer mind transfer to occur in the twenty-first century, as is being predicted, such events must happen within the next hundred years. However, consider why these events must occur at all for human-computer mind transfer to work. The basic idea is to put your mind into a robot brain, which will be some kind of compact computer. If the computer is just not capable of supporting your level of intelligence, then such a transfer is undesirable, it is not plausible to consider it a transfer of your "whole" mind, and therefore it may not even be a transfer of "you." The basic idea is to maintain or improve your lot, not turn you into a dolt that has little claim to being you anymore. Therefore, computers, or the robots containing them, will have to be at least as smart as humans are. But being a human person involves more than just being intelligent--it involves being conscious and having other features of personhood, such as exercising a free-will, or at least thinking we do, and being able to act morally. We cannot lose these features in the transfer if we are to remain persons, so another requirement is that robots must be capable of supporting personhood, or being persons. In addition, of course, there must be a viable mechanism of transfer for it to work. Finally, part of the inspiration for being interested in human-computer mind transfer is to gain immortality. It would certainly seem to be the case that for this to happen the robot being existing after the transfer must be the same person as the one who went into the transfer (as the human). Therefore, it seems that one's personal identity would have to be, maintained--transferring my mind to a robot will allow me to continue *my* existence as the robot.

The authors who write about such matters think that not only will human-computer mind transfer be possible in the next century, it will actually happen. We are not going to worry about this distinction between it being able to happen and it actually happening. I'm not interested in the question of whether if it might be technically possible to pull it off, say in 2090, political and economic factors will allow it to really happen or instead prevent it from happening in anything more than research labs. If during the next century, we have computers capable enough to allow it to happen, that is good enough for our purposes, whether or not political and social factors allow it to become a widespread phenomenon in society. We are interested in the possibility and will assume that if it is possible for it to happen, in the sense that the robots and transfer technology are ready and other technical and philosophical questions have been resolved, then that's enough to make the claims about it happening plausible.

Phillips makes some comments on terminology here. We are mainly interested in robots, which might be, considered embodied computers. The general idea is that the robot will have some sort of computer as the equivalent of a human brain, with some kind of body

incorporating different types of sensory apparatus. The body need not be, made of metal; as we will see; new technologies may involve other materials. (However, he will not consider a being whose body is an exact duplicate of a human, made of human tissue, to be a robot. The idea behind human-computer mind transfer involves more than just making a duplicate human being.) I will use the term “robot” most often but sometimes he will use the term “computer” when I mean the computer in the robot or the robot itself.

“Robot” is from the Czech word “robota,” referring to a peasant or someone engaged in forced labor. “Robot” was first used in a 1920 play to refer to mechanical creatures who could carry out routine tasks without much external instruction. Thus, the original use of the term refers to an “intelligent” machine that is mostly self-controlled. Associated with “robot” are the terms “android,” which refers to a bi-pedal robot that looks human, and “cyborg,” which is a combination of human and machine (Paul & Cox, 1996, p. 25) [1]. Mostly I just use the term “robot,” though it may be that the robots in question turn out to be androids. On some human-computer transfer scenarios involving transplants as part of the mechanism, the intermediate beings might be, considered cyborgs.

He also uses “computer” in the sense most people mean, for modern digital computers. In the future they might be made of different materials. On my usual use of “computer,” a human brain is not a computer. But some authors use the term in an extended sense for “any system that uses signals to process information via calculations that solve algorithms according to a set of rules (Paul & Cox, 1996, p. 38) [1].” In this extended sense of “computer,” it may be that the human brain is a computer. I do not use “computer” in the sense Searle (1981) [2] does, in which everything can be considered a computer because it can be interpreted as instantiating some function or algorithm.

10.3. Sources Describing the Extraordinary Future

The basic position this thesis will discuss holds that within the next century computers and robots will be at least as smart as humans will, robots will be conscious persons with synthetic bodies, and human beings will be able to continue their personal existence by transferring their minds to such robots. This may seem a startling claim more at home in science fiction than science, but to scientists holding this view the position is, considered between the two in the realm of “science speculation.” We will borrow a phrase from Paul and Cox (1996) [1] and henceforth refer to the happening of this series of events as the “extraordinary future.”

Apparently, many computer scientists and private industry computer executives working in artificial intelligence believe that eventually we will develop robots as smart if not smarter than human beings. This view is not usually, discussed in scientific journals but comes out in interviews, popular articles, etc. Of course, not all of these scientists and executives believe, as the authors we will examine do, that this will happen during the next century.

Some of the scientists who believe in the upcoming era of robot intelligence also hold that such robots will be conscious persons. Probably still fewer believe that we will be able to transfer our minds and identities to such robots, but among those who do are a few who make this prediction in some of their popular writings. For my description of the extraordinary future, we rely on Ray Kurzweil, Hans Moravec, Gregory Paul, and Earl Cox. Kurzweil,

founder and executive of several companies involved in pattern recognition and other artificial intelligence, presents this idea in his recent work, The Age of Spiritual Machines: When Computers Exceed Human Intelligence (1999). Moravec, founder of and now Principal Research Scientist at the robotics program at Carnegie Mellon University, presents this position most recently in Robot: Mere Machine to Transcendent Mind (1999) but also earlier in Mind Children: The Future of Robot and Human Intelligence (1988). Paul, an evolutionary biologist, and Cox, a computer industry executive, present their ideas in Beyond Humanity: Cyberevolution and Future Minds (1996). Throughout this chapter, we will use the phrase “our authors” or “the authors” to refer collectively to this group of individuals.

For my depiction of the extraordinary future, we really will draw on just the above authors. But as mentioned, others have similar opinions. We will not draw on the comments of these other thinkers because it might not be considered fair to critically analyze a position on the basis of some casual comments made in a brief interview, for example. However, the reader interested in hearing such comments should consult Fjermedal's The Tomorrow Makers (1986). Fjermedal elicits a variety of comments sympathetic to the extraordinary future from distinguished computer scientists. Moravec is, quoted extensively. Nevertheless, there are others. For example, Marvin Minsky apparently believes human-computer mind transfer will happen, though not in his lifetime. Danny Hillis sees it coming, as does Gerald Sussman, who thinks it could happen in the next generation (Fjermedal, 1986, pp. 7-8). Such thinkers are well, known in the computer field.

In each of the following chapters of this write up, we examine one of the basic predictions Dr Phillips laid out above, namely, that robots will be as smart as humans, that they will capable of being persons, that there will be a viable mind transfer mechanism, and that human-computer mind transfer will allow maintenance of personal identity. In each section of this chapter, we proceed first by describing the pertinent views of our authors, mentioned above, after which Dr Phillips is bring in supplementary material and critically discuss the plausibility of the prediction coming true. It may help you to follow what is going on later if at the start here I lay out the extraordinary future in a little more detail, so below we flesh out my earlier description. Discussion in more extensive detail and citation of specific sources will have to wait until later chapters.

10.4. DESCRIPTION OF THE EXTRAORDINARY FUTURE

Our authors think that in the next century computers will get very smart. The reason is that computing power will continue to increase exponentially. Within the next century computing power available in a compact package will first equal the computing power of the human brain and then far surpass it.

Several approaches are, used to estimate the computing power of the human brain. One method is to try to base it on the number of neurons, connections, and firings in the brain. The human brain has roughly on the order of a hundred billion neurons, with each neuron connected to as many as thousands of others. Each of these connections is a synapse, so there are trillions of synapses. An estimate is then, made based on taking each of these connections to represent a calculation and noting the number of times neurons can fire a second. Similarly, the number of brain connections is, used to estimate the memory capacity of the brain.

Another method for estimating the computing power of the brain is to first estimate how much computing power it takes a digital computer to carry out a particular brain function, and from this extrapolate to how much computing power it would take to do all the brain functions.

Estimates for brain computing power vary among our authors, but typically it winds up being in the hundreds of trillions or quadrillions of calculations or instructions per second, and memory is estimated at about the same number of bits.

Our authors also note that the brain appears to work by some kind of massive parallel processing rather than by the kind of serial or sequential processing typical of modern digital computers. While in the future computers will involve more-parallel processing than they do now, they should not have to match the massive parallelism of the human brain. This is because they can make up for less parallelism by running faster than the brain can.

Though modern computers far from possess this level of processing power now, computers in compact form should attain the requisite processing power sometime around 2020 to 2040. This is because, as Moore showed in 1965, processing power had to that point doubled every year. Moore also claimed that this trend would continue; computing power would grow at an exponential rate, doubling every one to two years. The claim that it will has become, known as “Moore's Law.” The authors think that this exponential rate of growth can be, sustained by the use of new technologies; these new technologies will also enable the creation of robot bodies to house these fast computers. Such technologies include nanotechnology, atomic (quantum) computing, transmutation of elements, and artificial life.

Clearly, then, according to our authors, robots will be very intelligent. Given that such intelligent compact computers in robot bodies will be available within the first half of the twenty-first century, the next question is one of whether they will be able to be conscious persons. If a human plans to transfer a mind into such a robot, the robot must be able to support the mind and other aspects needed for being a person.

One view of the mind that is consistently, rejected in the discussions of our authors is the view that the mind is a soul or substance distinct from the human brain. This view is, seen as an outmoded religious or superstitious position. To our authors, the mind is, seen more as the brain, various patterns in the brain, or the “program” running in the brain. The analogy here is with computer software. The mind's relation to the brain is like a program's relation to hardware running it. So in this respect robots should have no trouble having a mind, since the mind will be just the software running their robot brain hardware. What of consciousness and subjective experience? The view is that consciousness has a physical basis, so the claim is, made either that such robots will be conscious or that we will come to believe they are conscious because they will act as if they are. Robots will also exhibit emotions. Though they do not put it in these terms, our authors clearly believe that robots will have all essential aspects of personhood.

So we have these intelligent, conscious robots available--how exactly do we get our minds into them? There are several stages to this, and for each a number of scenarios are, presented. First, we have to find out what is in an individual's mind. Kurzweil notes that invasive (destructive) or noninvasive techniques might be used. Invasive procedures would include tearing apart the brain while observing its structure. Noninvasive techniques would include scanning, for example sophisticated MRI or optical scanning. Future MRI and optical scanning will be at the neuron or sub-neuron level and will allow observation of individual neurons firing. However, there might be other ways. Moravec suggests several possibilities

for computers attaching themselves to humans, plugging into electrical circuits in the brain, and learning brain details by observing the neural traffic.

The second stage of the transfer is to build the robot and transfer the mind to it. We have to either first build a robot brain, and then transfer the results of the scan into that existing robot brain or else do these two steps simultaneously. Most scenarios seem to follow the latter method; they depict not an already constructed robot awaiting a transfer but the creation of the robot brain equivalent to the human brain of the person transferring, and so the transfer seems to be the same process as the robot brain construction. Nevertheless, on a few scenarios we could break this second stage down into two distinct stages, with the second stage being the construction of the robot and the third stage being the transfer.

Our authors mention a lot of possibilities, and so under both schemes there may be multiple ways to accomplish the construction and transfer. Some depictions have the robot built to exactly copy the structure of the human brain of the person transferring, and when the robot is finished the mind has been effectively transferred. Other scenarios suggest the reverse engineering of the program or programs running the human brain. Once the program running on a particular person's brain has been determined, equivalent code can then be, written for a robot brain. This robot brain might already be built, and then programmed with the equivalent code, or it might be built and programmed at the same time. In either situation, the structure of the robot brain would not necessarily mimic that of the human (whether at a fine level of detail ("fine-grained") or a coarse level), but the overall functioning would be equivalent. Furthermore, on another variation, if the scheme is to build the robot brain "on the fly" as the human brain information is being obtained, either the robot brain could be built physically separate from the human brain or parts of the robot brain built and substituted for parts of the human brain in a series of transplants.

In general, the authors believe that personal identity will be preserved during the human-computer mind transfer. The consensus is that one's identity does not consist in having the same body or brain over time but in something else. It is a little difficult to say what that something else is, though. It is variously described as a continuity of "patterns of matter and energy," or of mental patterns, along with a continuity of memory. The basic idea is to preserve in the transfer whatever it is that gives each of us our sense of a self-conscious awareness of being the same person through time.

On the other hand, there are some signs of the realization that puzzles about identity might arise, though little attempt is made by our authors to address them, much less solve them. For example, if the transfer occurs by building a robot "brain equivalent" of my human brain through noninvasive scanning, then the robot could be fully functional while my human brain and body still are as well. The transfer has allegedly taken place by the time the robot was completed. Therefore, it seems we have two of me--the one with the robot body and the one with the human body. This might be seen as a difficulty, or at least a puzzle, concerning my identity--which one am I, possibly both? Our authors are aware of this possibility, and sometimes mention that we might experience puzzlement at it, but they do not provide a clear answer to the question raised. On the other hand, if my human body were, destroyed some time before the robot was created, it is not clear whether the robot would be me or some other new person. So timing might be important to whether "I" go on or not. But how much time would have to elapse between the destruction of me and the creation of the robot before it would not be me but a new person? A split second? A day? Two weeks, etc.? This is another

difficulty or puzzle, that clearly arises on some transfer scenarios but no clear answers are provided.

Moravec seems to think that you can keep various copies of yourself stashed around the universe, so that if one of them is, destroyed, you can still go on living as all the others. And if all are destroyed, other people or robots can pull out the blueprints and build you a new body and brain on the spot, with this body and brain really still the old you continuing to survive. Obviously in his discussion Moravec presupposes particular positions on a lot of controversial issues about personal identity. For example, it is certainly possible to wonder whether all these copies are really, just “relatives” of some sort rather than you.

This completes our preliminary sketch of the extraordinary future. In coming sections, we turn to consider the view in more detail and subject it to a more thorough examination.

10.5. Computer Intelligence in the Extraordinary Future

One requirement for the extraordinary future is that computers will be as smart as humans will. Actually, the authors who present the extraordinary future clearly think that within the next century computers will far surpass humans in intelligence. In this chapter, we describe their reasons for making this claim and consider whether it is plausible. In order to do this I have to consider related issues such as the nature of human intelligence, how the brain works, how computers work, realistic projections of increases in computer processing speed, and different understandings of the concept of thought.

Our authors think computers will get very smart, even smarter than humans are, during the next century. They approach the question of how smart that is by trying to estimate the computing power of the human brain. It is perhaps surprising that in considering the question of how smart humans are, our authors have very little to say about human intelligence. Rather they discuss how big and powerful the brain is because they tend to think of human intellectual performance in terms of the “computing power” of the brain.

To understand the computing power of the brain, they think, you need to understand a little bit about brain anatomy. Paul and Cox note [1] that the brain has about 10^{12} cells, but less than $2*10^{11}$ are neurons (other estimates, such as that from Kurzweil [3], say closer to 10^{11}). Axons and dendrites from the neurons connect to other neurons via gaps called “synapses.” The neurons send chemicals (or electro-chemical pulses) across the synapses. Paul and Cox point out that these pulse signals are both digital and analog. The digital aspect is that either they are on or off; the analog aspect is in the fact that the signals fluctuate in peaks and valleys. Pulses across synapses, then, can come in different strengths. The synapses store memories as chemical and structural changes that equate to changes in the strength of the connections (Paul & Cox, 1996, pp. 136-140) [1].

Paul and Cox estimate that a single neuron connects to 10^4 neurons, though typically other estimates are for about 10^3 connections per average neuron. Paul and Cox [1] estimate a total of about 10^{15} synapses for the brain (Paul & Cox, 1996, pp. 136-137) [1]. Our other authors estimate a total of about 10^{14} synapses for the brain neurons but these seem to be associated with an estimate of only 10^3 connections per neuron.

So estimates of the size of the brain (number of neurons, synapses, etc.) can be different (by a factor of ten, for instance). Estimates of the computing power of the human brain also

vary and there are even two approaches to estimating. One approach is based on size and the other approach is based on functions. On the first approach, the size of the brain is estimated (as above), along with the number of connections among neurons, neurons in use at one time, and neuron speed, and from these factors one then estimates the number of calculations the brain can perform (per unit time). The second method is to base the brainpower estimate on an estimate of the computer power needed to do the brain's functions. On this latter approach, one might consider a particular function of the brain, such as visual pattern recognition, and note how much computing power a computer takes to do this. Well, if a computer takes this much power to do this one function, and the brain can do X such functions, then how much computing power would a computer need to do these X functions? Our answer is the computing power of the brain.

Let us start with the first approach, based on the size of the brain. Modern digital computers have hardware both for processing and for memory/storage (for example, Random Access Memory (RAM)). The brain seems to have its memory/storage mixed in with its processing hardware, but processing and storage might be, considered distinct for purposes of estimating. Here is Kurzweil's reasoning. He estimates the human brain has about 10^{11} neurons. With each neuron having an average of 103 connections between it and other neurons, the total connections are about 10^{14}. (Note that these last two numbers are 10% of those of Paul and Cox). If each connection is, considered a calculation, this allows 10^{14} simultaneous calculations. (It is not, made clear why each connection is a calculation though.) Neurons can handle 200 calculations per second, so this gives $2*10^{16}$ total calculations a second. Kurzweil's estimate of the memory capacity is a little more puzzling. He says that the memory capacity of the brain is about 1014 total synapse strengths, but this is the same number he used for the total of synapses. This would mean each synapse has only, one, strength, when in reality each synapse has much strength. What he probably means is that there are 10^{14} total synapses, but the total strengths of all these synapses together should be represented by the equivalent of a 10^{15} bits, which implies about 10 strengths per synapse (with a bit for each strength). Therefore, Kurzweil arrives at an estimate of a 10^{15} bits' worth of total brain memory (Kurzweil, 1999, pp. 103-104) [3].

Moravec, is using the second method mentioned above, estimates brainpower by basing it on the computer power that would be, needed to accomplish the brain's functions. He considers how much computing power is required for visual pattern recognition, and then extrapolates this to how much computing power would be required for all that the brain can do. The brain, Moravec thinks, would have to operate at about 10^8 MIPS. Considering the memory required to give acceptable performance, he notes that one megabyte per one MIPS is a good rule of thumb. Therefore, the brain should have about 10^8 megabytes of memory. This is the same as Kurzweil's number of a 109megabits (10^{15} bits). Another way to see this is to note, as already mentioned, that the synapses' memory capacity comes from their ability to be in a number of distinct states or strengths through “molecular adjustment.” Moravec guesses that each synapse can be in a byte's worth of states, and this is close to what we surmise must be Kurzweil's conjecture of 10 bits. So 1014 synapses is equivalent to 1014 bytes of memory, which is the same as the 10^8 megabyte figure mentioned earlier (Moravec, 1999, pp. 54-56) [4].

We have already seen some variance in the size estimates of the brain, based on the estimator. Brain size can be 10^{11} to $2*10^{11}$ neurons; synapses per neuron can be 10^3 to 10^4; total synapses can be 10^{14} to 10^{15}. Kurzweil [3] puts the speed of a neuron at 200 firings a

second, Paul and Cox think it 100, and Moravec seems to think it can go up to 1000. Paul and Cox note [1] that at most only 10% of neurons are firing at any one time, which is not mentioned by other authors, which may not be relevant, since most estimates of computing power are based on the number of firings per second, and all neurons could fire repeatedly in one second without all of them firing at once. Paul and Cox note [1] that estimates of total brain speed range from 10^{13} to 10^{15} calculations a second. However we have already seen that Kurzweil estimates it at $2*10^{15}$ calculations a second (Paul & Cox, 1996, p. 139) [1] Both Kurzweil and Moravec hold to an estimate of memory in the range of 10^{14} bytes/10^{15} bits, which Paul and Cox agree with. However, Paul and Cox also note that this may underestimate memory, since the same synapse may be involved in many different remembered situations, with different sets of other neurons for each situation, for example (Paul & Cox, 1996, p. 140) [1].

Therefore, in terms of neurons and connections the brain is very large, but in terms of speed each neuron is comparatively slows, with neurons able to fire only about 100-200 cycles per second. Nerves themselves can manage to send signals at only about 100 meters a second. What makes the brain powerful is not neuron speed but the extensive parallel processing of billions of neurons involving trillions of connections (Paul & Cox, 1996, p. 139) [1]. Paul and Cox take it to be a massively parallel machine, since they define a machine as something “that uses energy to do work or that processes information according to an internally logical set of rules” (Paul & Cox, 1996, p. 140) [1].

Having some rough, idea of the computing power of the brain, our authors can tackle the issue of when it is that computers will be able to match and even exceed this computing power. There seem to be two main lines of argument used to attempt to show that this event will take place early in the next century. First, by examining advances since the turn of this century, proponents find that computing power is advancing at an exponential rate. This is, expressed in Moore's Law. According to our authors, this exponential growth rate can be safely, extrapolated into the future--proponents find no compelling reason to think that it will stop anytime soon. Moreover, advanced technologies for building such computers and for building robot bodies will be developed. This line of reasoning seems to be the primary argument for the claim that computers and robots will meet and exceed human intelligence during the next century, but sometimes a second argument for computer advancement appears, one that involves claims about the nature of evolution and the universe. More about this, second argument later.

Let us describe the first argument. The most widely known depiction of the growth of computing power is known as Moore's Law, after Gordon Moore, an inventor of the integrated circuit and a former chairman of Intel. It seems as if everybody these days goes around quoting Moore's Law, so it is ironic that there seems to be a little controversy over exactly what the law states! In 1965, Moore noted that the surface area of a transistor (when etched on an integrated circuit) was undergoing a 50% reduction of size every twelve months. This 50% size reduction correlates with a doubling in speed. In 1975 he was reported as revising that time period to eighteen months, but Kurzweil notes that Moore himself later claims he had revised it to twenty-four months! Whatever the exact rate, Kurzweil notes, many engineers currently believe that this growth rate cannot be sustained indefinitely and probably not beyond 2020 or even earlier. By that time the transistor insulators will be only a few atoms thick and further shrinking by conventional means will be impossible (Kurzweil, 1999, pp. 20-21) [4].

If Moore's Law is going to run out of steam in the next dozen to twenty years, how can our authors believe that computers will continue to advance well into the second half of the next century? Kurzweil points out that the exponential growth of computing did not begin with Intel or Moore's Law. Going back to early computing machines such as a 1900 Analytical Engine and a 1908 Hollerith tabulator, he claims that one can observe essentially exponential growth in computing speed per unit cost during the entire century. At the start of the century, the rate is one of doubling every three years, while now it is a doubling about every twelve months. The fact that the exponential growth of computing did not begin with Moore's Law suggests to Kurzweil that it will not stop with the end of Moore's Law (Kurzweil, 1999, pp. 20-25) [4].

We have already seen that the exact rate of growth is not clear, though it is now commonly, placed at a doubling of speed every twelve months to two-years. For predictions about exactly where computing will be at various points in the next century, our authors for the most part just draw a graph showing a curve or line representing the growth and extrapolate that curve or line into the future. Since no one knows, exactly how limitations of current chip technology will be, overcome to pass the hurdle many envision for 2020 or earlier, none of the authors can prove that the exponential growth rate will continue or say exactly how it can be sustained. They do however suggest currently undeveloped technologies (such as nanotechnology and atomic computers) that they think might be, used to meet the expected challenges.

Let us turn to specific predictions. Moravec notes that current supercomputers can manage only a few Million Instruction Per Second (MIPS), whereas we have seen that to match the human brain we need 100 million MIPS. In Moravec's sketch of future robot generations to be developed during the next century, fourth generation robots will arrive in 2040 and have a processing power of 100 Million Instruction Per Second (MIPS). A major feature of these robots that will distinguish them from earlier robots is that they will be able to reason. They will be able to simultaneously, simulate the world and reason about the simulation, and they will understand natural languages (Moravec, 1999, pp. 108-110). Therefore, Moravec thinks small computers will match human brainpower by 2040.

Kurzweil claims that computer speed at the turn of the century doubled every three years, then every two years in the fifties and sixties, and now every twelve months (Kurzweil, 1999, pp. 2-3). In 1997, $2000 of neural computer chips would get us $2*10^9$ calculations per second, but by the year 2020 this will have doubled 23 times and reach $2*10^{16}$ calculations per second, which is in the range of the brain. Memory prices halve every eighteen months, and the requisite 10^{14} total synapse strengths should be available at the right price in 2023 or even sooner. Therefore, a $1000 personal computer in about 2020 should match the power of the human brain. (Supercomputers should be there even earlier, by 2010, but these are physically too big for a robot body). So by about 2020 small computers will be able to read and understand written documents; they will then gather knowledge on their own by reading, and so learn both all human-acquired and machine-acquired knowledge. Doubling power per price every 12 months, a personal computer will match a small village by 2030, the population of the US by 2048, and a trillion human brains by 2060 (Kurzweil, 1999, pp. 4-5, 103-104) [3].

Paul and Cox estimate the human brain processing speed to be from 10 to 1000 teraflops (10^{12} calculations per second). We are already at the 1 teraflop level, and they estimate that it should take 20 to 25 years to build a petaflop (10^{15}) machine the size of a mainframe, and

about 35 to 45 years for this to be in a small computer. Small ten teraflop machines should be available by 2020. Memory requirements (internal RAM type) might be about 1015 bits. Computer memory capacity improves at about the same pace as processing speeds, so the memories of powerful large computers should reach the human range at about 2015, with two decades later this much available in small computers. Memory space on peripherals will move from magnetic tape or disks to something three-dimensional, like the brain, preferably in a holographic manner (Paul & Cox, 1996, pp. 203-206).

We earlier saw that estimates of brain size, brainpower, and brain memory varied, and now we have seen similar variation in estimates of when computers will match the computing power of the brain. Kurzweil may be more optimistic than Moravec, Paul, and Cox, but all the authors would agree that by 2040 there would exist robots whose computer brains are the equal of the human brain. After that, the computing power of robots will grow to quickly far exceed that of humans.

Note that in these estimates there is no insistence that these robots be massively parallel processors like the human brain. The concern is with sheer computing power. The prediction is only that the computers in question will match the overall power of the human brain in terms of total calculations or instructions per second. Because of this, the assumption seems to be, the two will be equivalent in terms of function.

In the coming century and beyond, how will such technological feats be, accomplished? Paul and Cox claim that when smaller chips of silicon reach their limit, “top-down silicon etching” will be succeeded by “bottom-up nanotechnology” (Paul & Cox, 1996, p. 210) [1]. Kurzweil thinks that advances in chips may occur through incorporating a third dimension in chip design. Improvements in semiconductor materials (with superconducting circuits that do not generate heat) will allow chips with thousands of layers of circuitry. Combine this with smaller component geometries, and computing power will improve by many millions. Other technologies that may play a part include nanotube, optical, crystalline, DNA, and quantum (Kurzweil, 1999, pp. 33-35) [3].

Beyond the issue of chip advances, the breakthroughs needed to create intelligent robots will come from combining advances in artificial intelligence, artificial life, the aforementioned nanotechnology and transmutation of elements into other elements. Each of these elements alone will experience limitations. Artificial intelligence will find it increasingly difficult to carry out the extreme complexity involved in programming high-level intelligence (long since beyond simple human programmers). Artificial life can evolve to high complexity, but so far, it is inside computers that are not of high enough complexity. Nanotechnology will be able to build complex hardware, but it will be dumb. Transmutation will produce raw exotic materials, but it cannot put them together. But connections among all these technologies will enable them to work together. Artificial life will be the means by which artificial intelligence can evolve beyond the limits of human programmers. Combining nanotechnology and artificial life will allow growing the complex computers to run at a high intelligence. Heavy elements on stellar scales could be, provided by transmutation (Paul & Cox, 1996, pp. 124-125) [1].

Above I mentioned that the concern about advances in future computers was more with overall processing speed than with building into them massive parallelism. Nevertheless, there is acknowledgement among our authors that some parallelism might be, needed. To think like a brain, it might be that computers have to work somewhat like a brain. They will have to have complex parallel processors running many algorithms at once. Programming

these computers will be incredibly complex, and as mentioned a combination of conventional programming (using human and artificial intelligence) and artificial life may succeed. There is also agreement that such a computer will have to be a self-learning neural network storing holographic memories by strengthening and weakening connections. But these computers may not have to be as complex as the brain. Brains have more than a hundred billion neurons and trillions of interconnections because the neurons are so slow. A faster computer of a million cycles a second could use only millions of neural circuits and match the total speed of the brain (Paul & Cox, 1996, pp. 216-220) [1]. Note here the implicit recognition that the real target of robot brain development is the matching of the total computing power of the human brain. The assumption seems to be that the human brain's primary reason for being so parallel is that human wetware is so slow, not because massive parallelism has other needed virtues.

Now, for the second argument for computer advancement that, we mentioned earlier. The proponent here is Kurzweil, who in this second argument makes sweeping claims about the nature of evolution. He draws upon several laws or principles he believes hold for evolutionary processes and other processes. Moore's Law is not just a set of industry expectations but part of a deeper phenomenon. The “Law of Time and Chaos” holds that in a process such as evolution, the time interval between critical events (that significantly affect the future of the process) increases or decreases with the amount of chaos. Chaos is the quantity of disordered, random events relevant to the process. Order increases as chaos decreases. So a sub-law of the above law is the “Law of Accelerating Returns,” which is that as order increases exponentially, the time interval between critical events grows shorter (or as Kurzweil puts it, “time exponentially speeds up”) (Kurzweil, 1999, pp. 27-30) [3]. In accordance with the “Law of Accelerating Returns,” when Moore's Law gives out by 2020 another computational technology will have arrived.

As one can see from the above description the authors in question think that it is clear that humans will soon be matched and then quickly outclassed by robot intelligence. At this point, we would like to turn from exposition and description to evaluation and appraisal. I think our authors are overly optimistic in their estimate of how easy and how soon robot intelligence will come. In fact, I think there may be such severe problems in getting a robot to be as smart as a human that we can put no realistic timetable on when it will happen (if ever). To support this less than rosy appraisal I will suggest the following:

1) While it is true that computing power has advanced, we cannot say with any confidence that this will continue at any particular rate. See “Advances in Computing Power” below.
2) Reliance on future technologies as the magic bullet is highly speculative. See “Reliance on Future Technologies” below.
3) While our authors realize that the brain appears to operate via something like a massive parallelism rather than by sequential processing, they do not fully realize that this may make the need for faster computers less relevant than the proper style of processing. See “Computing Power Alone May Not Be the Answer” below.
4) It may be a mistake to characterize humans primarily in terms of the “computing power” of the brain rather than in terms of intelligence. It may be very difficult to get a computer to display humanlike intelligence. See “The Many Aspects of Intelligence” below.

5) The Turing Test is not an adequate test of intelligence or adequate for purposes of evaluating robots for human-computer mind transfer. See “The Turing Test” below.
6) We may not be able to do anything more than merely guess that a computer can really understand anything. See “Computers and Understanding” below.

10.6. ADVANCES IN COMPUTING POWER

Claims about computing power advancing at exponential type rates are argued in two ways by the authors in question. First, there is the extrapolated line approach that graphs computing density, speed, or some such proxy for computing power over an extended historical period and projects the line or curve into the future. The other approach is, represented by the second approach of Kurzweil, who claims to discern fundamental laws of the universe about evolution, chaos, and order that shows computing power must advance and in an exponential fashion. This second approach is seen to even encompass Moore's Law, which of makes up a major part of the first approach.

Let us quickly tackle the second approach first. Kurzweil claims to discern such laws, but it's hard to take this claim seriously. Kurzweil's a smart guy, but here he comes off as discerning fundamental rules of the universe that almost no one else can see. Obviously, maybe he is the only one that can see them because they are mostly imaginative speculation. As far as I am aware, the laws he claims to discern are not considered established scientific laws, in the manner for example of the laws of Newtonian or quantum physics accepted by the majority of reputable physicists. Kurzweil does not provide any decisive, or what one would consider, even substantial evidence for their truth. They seem to be rather part of Kurzweil's personal metaphysical or even religious views. As such, while granting their intelligibility for the sake of argument, we cannot see that they constitute legitimate support for predictions of the extraordinary future any more than other religious views might be, thought to do so. For all I know, or anyone else does, these “laws” might be true, and they might be false. However, since insufficient evidence to establish their truth has been, presented, they do not give any real support to other claims about advancing computer intelligence. The first line of argument must provide any plausibility for the claim that computer performance will continue to increase at a particular exponential rate instead.

We will not attempt to analyze the truth of the claim that computing power has been proceeding at a particular rate for the last hundred years or longer because I think the notion of computing power applying to time periods prior to the development of digital computers may be suspect or at least hard to measure. Furthermore, while our authors see Moore's Law as only part of a wider phenomenon, Moore's Law is as clearly, defined, as this rate is likely to get, so a defense of Moore's Law is probably the best chance anyone has of making the case for this whole extrapolation approach. So how well does Moore's Law hold up?

Just about, everyone in computing knows about Moore's Law. Many in the computer industry, not just our authors, frequently cite Moore's Law. I have great respect for Gordon Moore, but when we take, a look at the much-ballyhooed Moore's Law, it turns out that the marketing hype has gone beyond the reality. What was originally an insightful (and perhaps lucky) prediction on the part of Moore has been blown up into something perhaps far beyond his original intent. It is not even clear what Moore's Law is anymore.

The original prediction of Gordon Moore was, made in a short article in a magazine called Electronics. The article was, called “Cramming more components onto integrated circuits (Moore, 1965).” At the time, Moore was director of research and development laboratories for Fairchild Semiconductor, and he had been, asked to predict the next ten years in the semiconductor industry (Schaller, 1997). In this article, Moore was concerned to emphasize that integrated electronics, as opposed to using discrete components, would be, increasingly used in all sorts of electronic devices because it offered a number of advantages, including reduced cost, increased reliability, and increased performance. Integrated electronics would be, increasingly used in computers to soon enable home computers. With respect to semiconductors used in computers, increased integration meant placing more components on a single chip (on a single integrated circuit) (Moore, 1965, pp. 114-115) [5].

Given that the number of components on a chip was increasing, Moore then tried to predict at what rate this increase was occurring and would continue to occur. We do not want to take away anything from Moore's insight here, but apparently he was not the first or only one to think that computing was advancing at a fast pace. It has been claimed that already by the mid-1960's there existed in the semiconductor industry the general understanding that innovation was proceeding exponentially; it turns out that Moore was just the most articulate spokesman for this position (Schaller, 1996) [6].

Note that Moore decided to discuss the state of the art for semiconductor integration in terms of the number of components that should be “crammed” onto a chip for the minimum average per component cost. Moore's discussion is in terms of “device yields” but I think I can paraphrase it in terms of the performance increase brought by adding a component to the circuit, though Moore clearly does not phrase it explicitly in terms of processing speed performance. When adding components to a chip, initially each component added lowers the average “per component” cost. That is, initially the performance added to the circuit by the additional component is relatively large compared to the additional cost to add that component. In terms of an analogy with what we might have learned in an economics course, during this phase the marginal revenue exceeds the marginal cost. Eventually, though, adding more components actually starts to raise the per component cost because the performance each additional component adds to the chip becomes relatively small compared to the additional cost of adding that component. To return to the economics analogy, the marginal cost starts exceeding the marginal revenue. Now, the crucial point is that for any given time in the development of technology (any given date in time), there will be an optimal number of components that can be placed on a chip to give the minimum average component cost. In terms of our analogy, this “optimal number” would be analogous to that point at which the marginal revenue equals the marginal cost. At the time, Moore wrote, in 1965, the minimum average component cost was, reached at about 50 components per circuit, but Moore could see that the number of components at which this minimum average cost was reached was rising as technology advanced (Moore, 1965, p 115) [5]. For example, it would cost less per component to produce a 50-component chip in 1966, than it did per component to produce a 50-component chip in 1965. However, more importantly, because production technology had advanced, in 1966 the lowest average cost per component would no longer be, found on a 50-component chip but on a 100-component chip. Note also that Moore clearly had production chips in mind, not just special expensive prototypes that would never go into production, because his discussion is in terms of the complexity the chip should have for the minimum average component manufacturing cost (Moore, 1965, p. 115) [5].

To make an estimate of the future rate of increase in the number of components for minimum average component manufacturing cost per chip, Moore would extrapolate from what he saw as the then current trend. To determine that then, current trend and Moore chose five data points. The first data point was the 1959 production of the first planar transistor. The second, third, and fourth data points together comprised what was really more like a set of data points that represented the first few integrated circuits of the early 1960's, including the 1964 production of IC's with 32 components. The last data point was the soon to be released (in 1965) IC with 64 components. Moore plotted the points on logarithmic paper and connected them with a straight line, and then extended the line to 1975. Reading from the line, which shows a doubling every year, he predicted that a chip would attain 65,000 components by 1975 (Schaller, 1997) [7]. As Moore put it, "The complexity for minimum component costs has, increased at a rate of roughly a factor of two per year" (Moore, 1965, p. 115) [5].

A little background will help us understand what was happening technologically at this time. Miniaturization was a major theme, as it is today. William Shockly and his colleagues at Bell Labs had invented the transistor ("transfer resistor") in 1947. The transistor was, based on the discovery that by adding impurities to a solid such as silicon the flow of electricity through it could be controlled. The transistor could be, made smaller, more reliable, and less power-hungry than the vacuum tube it would replace. By the late 1950's engineers at Fairchild had developed the transistor in a plane (a planar transistor), which Moore would later say was the beginning of the law of density doubling. Later came the first planar integrated circuit, which enabled the extending of the cost and operating benefits of transistors to mass-produced electronic circuits (Schaller, 1997) [8].

As semiconductors developed, advances in production technologies were as much an influence as technological inventions such as transistors and integrated circuits. For example, the development of a diffusion and oxide-masking process enabled the diffusion of impurities (dopants) directly into the semi-conductor surface, eliminating the tedious process of adding conducting and insulating material layers on top of a substrate. Sophisticated photographic techniques then enabled the laying of intricate patterns in the semiconductor so that only desired areas lay open to dopants. Production reliability and accuracy increased and the process moved from individual fabrication to batch production. Another production technology was the planar process itself, which by replacing three-dimensional transistors with flat surface planar transistors, made them easier to make and make smaller. Because the electrical connections were now, flattened, they could be, made by evaporating metal film onto the semiconductor wafer in appropriate areas instead of having to make them by hand. A photolithographic process was, used to etch regions on the chip, which were plated and laid on top of one another on the silicon wafer. Circuits could be, integrated with one another on a single substrate because electrical circuits were now internal to the chip (Schaller, 1997) [7]. We point out the significance of production technology advances because, when you are talking of not what goes on in the lab but of the kind of production chips needed for the extraordinary future, production advances are as much a part in advancing chip densities as any other technological breakthroughs.

Moore's 1965 [9] prediction for 1975 turned out to be accurate; in a 1975 paper presented at an IEEE meeting, Moore noted a memory chip of the proper density was in production at Intel, where Moore was now President and CEO. (The original article talked of memory density, which has not always increased at the exact same rate as microprocessor density,

though the discussion is often loose enough that authors perhaps unfortunately shift back and forth between the two types of chip.) Moore claimed that up until 1975 there were three key reasons for the rate of growth. First, due to the use of optical projection in place of contact printing for the lithography masks on wafers, they could make bigger chips with fewer defects. Second, they had ever-finer rendering of images and line widths. Third, “circuit device cleverness” enabled manufacturers to use more of the total wafer area. But this third device, “circuit device cleverness,” was ending with the Charge Coupled Device (CCD) for which a new technique of “doping” semiconductors used controlled light beams rather than chemical means. Therefore, going forward, the industry would have to rely on only the first two factors, bigger dice and finer dimensions (Schaller, 1997) [7]. In other words, to get more transistors on chips, they would have to resort to making the chips bigger and shrinking the lines used on the chips so that the transistors would be closer together.

So now, Moore wished to make a change to his prediction of the rate of future growth going forward. No longer, would it be a doubling every 12 months; it would have to be slower. However, here is where some controversy enters, as Kurzweil [11] notes. It is clear that Moore redrew his line from 1975 onward to have a gentler slope (Schaller, 1997) [7]. Nevertheless, there is some controversy over whether Moore changed the doubling period to 18 months or 24 months. We can find no record of publication of the paper he read at the conference, and later accounts of what Moore claimed do seem to vary. There are reports that Moore himself later claimed he changed it in 1975 to 24 months, but most observers instead report he changed it to 18 months (Rosenberg, 1999) [10]. One might guess, as some have, that the 18 months figure is really a conflation of the 12 months and 24 months numbers. But Schaller (1996) [8] claims Moore presented his law mathematically at the 1975 conference as “(Circuits per chip) = 2**((year-1975)/1.5),” which is based on an 18month period.

Later still, in 1997, Moore himself referred to the 30 year compounded growth rate as a doubling every 18 months, thus blending the two separate rates into one (Moore, 1997). Therefore, Kurzweil is entirely correct to point out that there is controversy over exactly what rate Moore changed his law to in 1975. In any event, shortly after the 1975 conference the prediction was called “Moore's Law” (Schaller, 1997) [7].

Therefore, our first lesson is that there is no one rate called “Moore's Law.” It was originally one rate, and then changed by Moore to another (and we're not even sure what that new rate was). Some even claim that it has changed again. Kurzweil [11], for instance, claims that Moore's Law has now picked up to a rate of doubling every twelve months, which I do not see supported by anything Moore says or by the majority of industry pundits. Moore himself slowed down the rate in 1975 rather than sped it up. Sure, you could claim the real rate was some kind of average of the two earlier figures, but to do this you would have to fudge the data points. Remember that we are supposed to be talking of a rate that holds constant over its life, not just an average applied to a large time period but not useful for predicting particular intermediate points.

Confusion over which rate to use prefigures the larger question of whether the rate is really all that exact anyway, even for a limited time period, because there is a lot of ambiguity about what the subject is. Are we talking of microprocessor chips used for CPU's or memory? Are we talking of chip density for chips already in production or for those just about to enter production? A few months' difference between the two could change the rate. Are we talking of chips available at any price or chips available at a competitive price (Moore seemed to mean the latter since his original discussion is in terms of manufacturing costs). Are we even

still talking of chip density? Note that chip density as used in this context seems to mean transistors per chip and not transistors per unit area of chip. When you double the total transistors by doubling the chip area, the chip density in this sense of "density" has doubled, rather than remained constant. However, some versions of the law claim it really is about a more traditional "units per area" kind of density. (For example, the Web-based "jargon dictionary," in an entry from an unknown author, states that Moore's Law holds that the "logic density" of silicon integrated circuits has followed the curve

$$\text{bits per square inch} = 2^{(t-1962)} \qquad \text{Eq. 7-1}$$

where t is current year (Moore's Law, 1995, italics mine) [12].

Generally, chip performance in terms of processing speed will improve with density increases gained by smaller dimensions, but chip size (the other way to gain "density") may not improve speed to the same degree. Moore usually talked of chip density, or the number of components on a chip, especially with respect to that needed to attain the minimum average component manufacturing cost, but later versions of the law often prefer vaguer terms such as "complexity," "speed," or "performance." This might have, been expected; Moore's Law does seem less relevant if chip speed, performance, etc. do not track increases in chip density. Even Moore was interested in performance of course; in a 1997 update, he noted the speed gains from smaller chips and shorter distances, something like a 20,000 fold increase from the 1971 4004 chip to the 1997 Pentium (Moore, 1997) [13].

One guesses that most of the variation in rate calculations and reporting must come from honest mistakes. Of course "fudging" the data to make it fit the rate is possible too. In the 1997 IEEE Spectrum article by Schaller cited above there appeared a chart showing rates for CPU microprocessor and memory chip density increases. However, after scrutinizing the data points on the chart, an observant reader pointed out that while the article and chart caption claimed a doubling every 18 months. The actual data points supported a different rate (a doubling every 22 months for Dynamic Random Access Memory (DRAM) and a little over 24 months for Intel microprocessors)! This reader even charged Intel with "revisionist history" on its Web site because the Web site allegedly claimed that in 1965 Moore predicted doubling every two years (which he did not) (Kane, 1997) [14]. One wonders: if Intel cannot get it right, then who can?

Currently (as of 1999) on Intel's official Web site, they provide a description of Moore's Law that is safely vaguer. They imply that Moore's Law is the claim that computing power, or capacity, would rise exponentially over relatively brief periods of time, roughly twice as much over a period of eighteen to twenty-four months. Intel claims the law is still accurate: in 26 years the number of transistors on a chip has increased more than 3200 times, from 2300 on the 4004 in 1971 to 7.5 million on the Pentium II processor of 1998 (Intel Corporation, 1999).

If one interprets Moore's Law as the claim merely that computing power will increase exponentially, with no time period specified, then the claim becomes so vague as to be almost meaningless. Certainly Moore did not understand it as vague in this fashion. He would hardly have found it necessary to "correct" the rate in 1975 if his claim were only that computing power would increase exponentially over some time interval or other.

So what is the status of Moore's Law, assuming we can agree on a version of what it is? It is hardly a scientific law or theory like Newton's “force equals mass times acceleration,” for instance. For one, it is not a law just about nature but involves contingent events and decisions in the business of making production chips. I have detailed above the particular technological breakthroughs, both production and otherwise, that enabled chips to advance at the rate they did during the sixties and seventies, and these breakthroughs did not happen by accident but by devoting time and financial resources to research projects and deciding to build plants to produce chips. The breakthroughs would not have happened without particular decisions to carry out the research, and the plants would not have been, built without particular business decisions to go after profits. There is no guarantee similar decisions will be, made in the future or that similar breakthroughs will occur even if the effort is, expended. We emphasize that keeping up with Moore's Law depends not just on advances in chip technology and production technology but also on the decision and financial ability of companies in free market systems to afford incorporating advances in this technology into the production of computer chips. If a broad recession or depression were to hit, as in the 1930's, it is not clear that computer companies would continue churning out advanced chips just to keep up with the law.

There have been admissions (even by Moore himself) that the law has become self-fulfilling. Moore pointed out that once the law became publicized and accepted, it became “more or less a self-fulfilling prophecy. The Semiconductor Industry Association puts out a technology roadmap, which continues this generation [turnover] every three years. Everyone in the industry recognizes that if you do not stay on essentially that curve and it will fall behind. So it sort of drives itself” (Schaller, 1997) [7]. Of course Moore's Law seems to hold, this line of argument claims, because companies put just that much effort into chip technology and production needed to meet what the law predicts they should have next year. It is as much a business target or objective as it is a “law.” For instance, a recent article by Meieran, of Intel, allowed that the industry is expending “enormous resources” to meet the predictions of Moore's Law (Meieran, 1998) [15]. Imagine the absurdity of claiming that the industry is expending tremendous resources to make sure that on the macroscopic level “force equals mass times acceleration” continues to hold in the future, and you see how far Moore's Law is from being a scientific law.

Of course, one of the things that drives the need for greater density is the need for the greater processing speed brought with it, and one of the things that drives this is the increasing size and complexity of software. Nathan Myhrvold, of Microsoft, points out that the Basic language had 4000 lines of code in 1975 and about half a million two decades later. Microsoft Word had 27,000 lines of code in its first version but by 1995 had grown to two million. In a process of reciprocal reinforcement, software makers consume new microprocessor capability as soon as the chip makers make it available, encouraging chip makers to continue increasing speed (Schaller, 1997). George Gilder claims that Bill Gates follows the dictum “waste transistors”; “every time Andy Grove makes a faster chip, Bill Gates uses all of it” (Schaller, 1996) [8].

The above considerations demonstrate to my satisfaction that it is difficult to find enough precision, objectivity, and grounding in science in Moore's Law to use it as a basis for anything more than a rough conjecture about the future. However, let us suppose for the sake of argument that there is some interpretation of Moore's Law such that the law has proven precise and accurate. Assuming Moore's Law has been precise and correct so far, what then

bodes for the future? Should we just confidently draw the line out through the next century, as our authors would have us do?

When we go beyond the comments of our authors, we find that opinions run the gamut. Pessimists believe the law is running on borrowed time and will soon play out. Some even believe we have failed to keep up with it already. Optimists believe that it will continue a long time into the next century (obviously believers in the extraordinary future are in this camp). Some extreme optimists claim we have advanced beyond it. (Note that this would falsify the law as much as if we had failed to keep up with it.) Given that the law itself appears in slightly different forms, one might expect some disagreement, but the range of predictions is so diverse as to indicate a real difference of opinion about the future of the law.

Let us talk about the pessimists first. Keep in mind that authors such as Kurzweil, Moravec, and Paul & Cox believe that computing power will continue to increase approximately in accordance with Moore's Law well into and past the middle of the 21st century. In this context, any estimate that Moore's Law will end significantly before this time can thus be, considered pessimistic. Because of this stringent standard, there are far more pessimists than optimists.

There have been claims that we have fallen behind Moore's Law already. For example, one article claimed Intel was actually falling behind Moore's Law in the introduction of chip improvements with the planned P7 introduction, around the millennium. Moore's Law predicted 170 million transistors per chip when only 10 million would be attainable (Rosenberg, 1999) [16].

Going forward, those who think time is running out for Moore's Law cite technological reasons or economic/business reasons. The technological reasons have to do with problems in making chips ever smaller. Economic/business reasons involve the falling price of chips and the increasing cost of chip plants.

Moore himself comes off as a pessimist, because he seems to think his law will be defunct within two decades. In a 1997 speech, Moore told his audience that his law was going to come into conflict with the laws of nature, namely the finite size of atomic particles, by about 2017. As chip production processes get smaller, more transistors can be, put onto a chip, offering the addition of new performance features. Since the distance between transistors is, reduced, the speed increases. Intel is currently (as of 1997) using a .35-micron process, will be moving to a .25-micron process, and then will go to .18 microns for 1000-MHz machines. (As of the date of this write-up, we are almost at the 1000-MHz machine level.) This latter process doubles the size of the processor and takes 40 watts of power, which generates problematic amounts of heat, so voltage levels would have to be, reduced from 3.3 volts to half a volt, which is “not fun.” Moving functions that are now off the chip (such as modems, graphics chips, and memory control) onto the chip attracts the interest of the Federal Trade Commission in possible anticompetitive practices (Kanellos, 1999) [18]. Then there is the issue of the cost of chip plants. In 1995, Moore pointed out those capital requirements for fabrication plants rise exponentially along with component densities. In 1966, a new fabrication plant cost $14 million, by 1995, it cost $1.5 billion, and by 1998 it would cost about $3 billion (Schaller, 1997) [7]. Moreover, costs of plants keep increasing; chips that are more complex require great capital to build plants, up to $4 billion per plant for the .18-micron chips, and it's not clear that rivals such as AMD will be able to keep up (Kanellos, 1999) [17]. What is sometimes, called “Moore's Second Law” is his claim that the cost of semiconductor plants doubles every three to four years (Geppert, 1998) [19].

Nevertheless, we may not even make it to 2017. In an announcement on June 25, 1999, Yahoo news reported that Bell Labs had announced in Nature magazine that they estimated chips would run into the wall in 2012. The problem would be the insulating material known as the gate oxide, which is the smallest feature on the chip. As the chips shrink, this has to shrink, and since it is already the smallest it can be, it becomes the limiting factor as chips use the .06-micron process, at which point it will be 5 atoms thick. New materials would be, needed to go any farther. However, the article also noted that 2012 would actually be an extension beyond the 2005 or so estimate of some previous scientists (Lemos, 1999) [20]. One of these other scientists was Gordon Bell, who actually predicts Moore's line will flatten out around 2003 (Geppert, 1998) [19].

So there are a large number of pessimists. A 1996 poll of 11 industry executives had allowed Moore's Law an average of only 14 more years--until 2010. Several reasons were, cited by these executives. For example, optical lithography may not be able to provide increasing density in a cost-effective fashion. Transistors could become so cheap that there would be no profit in making them smaller instead of just using more. As already mentioned above, it may become too costly to make them faster. Such cost increases are not a problem when chip improvements come even faster, but this is not, expected to continue. Given a time-period in which costs double, chip improvements are not expected to double (Schaller, 1997) [7]. Even if these numbers have not been, adjusted for inflation, the point is well, taken that continued increases in the speed and density of production chips may prove too costly to implement to allow Moore's Law to continue. Another problem cited is the anticipated difficulty of testing integrated circuits with many millions of gates (Geppert, 1998) [19].

The executives and others who claim that Moore's Law will be, maintained until 2010 or so might ordinarily be, interpreted as optimists, because they think it will at least continue for a while, but in our context, they are pessimists. It is truly rare to find anyone other than our authors who really expects Moore's Law to continue past the middle of the next century. One who is almost as optimistic is Nathan Myrvold, quoted above, who claimed in 1997 that he thought it would last for another 40 years (Geppert, 1998) [19]. Nevertheless, even this is more pessimistic than our authors are.

Of course, whenever a breakthrough is announced, there may be a few optimistic comments appearing that claim we are already ahead of the pace Moore's Law dictates. For example, in late 1997 Intel announced that a technological breakthrough would soon enable engineers to put twice as much, information (moving to 2 bit in the space of 1 bit) on flash memory chips. Apparently at this time Intel then announced something to the effect that Moore's Law was over--not because of technology pooping out but because the rate was going to be beaten (Geppert, 1998)! Another story in 1997 told of how IBM engineers had found a way to substitute copper for aluminum in chip circuitry, which would allow greater miniaturization. Copper is better than aluminum at the “below .20 micron” levels needed for new chips. (IBM would soon put this in play on their chips with Intel following by 2002) (Rupley, 1997) [21]. These news stories were even sometimes, interpreted as showing that this meant a return to the original 12-month period for Moore's Law (Rosenberg, 1999) [16]. Of course, since the rate itself is not precise, the same event may to one person, be the signal of a return to an earlier version and to another person the signal that the rate has been, beaten.

In response to pessimism about chip costs, and in line with the self-fulfilling feature of the law, one possibility suggested for semiconductor manufacturers is to team up with customers, competitors, suppliers, or even governments to share construction and R&D costs.

Advanced DRAMs were, developed by a joint effort of IBM, Siemens, and Toshiba. South Korea and Singapore enterprises appear to have state support (Schaller, 1997) [7]. Note the emphasis on the role of business to do its part to make Moore's Law happen, rather than just letting things play out and see if the law holds.

If Moore's Law can't provide the realistic expectation of a particular rate of computing power growth, then most likely the wider phenomenon of which it is a part will not do so either, since Moore's Law, vague as it is, seems stated more precisely and is better defended or justified than other parts of the phenomenon. Given the vagueness and pessimism we have seen above, however, we cannot have confidence in Moore's Law continuing far enough into the future to still hold at the time computer intelligence is alleged to match and surpass that of humans, roughly sometime between 2020 and 2050. It may or may not “give out” or “slow down” in a few years; we just do not know. (It is an interesting question, how much it would have to slow down to be, considered false. The fact that this notion is confusing is further evidence that it is not a conventional scientific law.) Moore's Law is not a scientific law in physics or chemistry but a self-fulfilling prophecy powered by economic and market forces as well as breakthroughs in technology. For any number of reasons, such as the limits of silicon circuit widths, the too-great expense of semiconductor plants, or the commoditization of computer chips and computers, the exponential rate of growth might change significantly, as Moore thinks it already did in 1975. I conclude that, as far as has been shown by our authors, Moore's Law or anything larger that is like it does not provide confidence that the requisite computing speed and power will be available when our authors predict it will.

It is interesting to consider in what sense the extraordinary future might be possible for the very lucky or the very rich even were Moore's Law to fail to hold. Recall that Moore's Law is about component densities for minimum average component manufacturing cost. Moore's Law has been generally interpreted as a yardstick of the computing speed and power available in the marketplace at an given point in time, which is in the direction of Moore's original understanding. But, strictly speaking, the original formulation of Moore's Law says nothing about the possibility that very dense or very fast laboratory prototype chips might be available that far surpass what could be produced in a manufacturing environment at minimum cost. This possibility might not seem likely, since when talking of reaching the limit of circuit widths at the 0.06-micron level, for instance, this seems a limit on chips per se and not just on minimum component cost chips. However, at some point in time there might be, developed an exotic chip technology that was too expensive to mass, produce but which was nevertheless possible to make in a research lab as a one-off prototype or the equivalent. So even was Moore's Law to fail to hold far enough into the future for low-priced robots to be available for the average person to engage in mind transfer. That failure would not preclude the possibility that high-priced prototype robots could be, made by research scientists, possibly for the very few rich who could afford maximum computing performance at any price. Thus, the failure of Moore's Law would not preclude the possibility of human-computer mind transfer becoming available for computer scientists working on such exotic robot brains or the few rich people who might be able to buy such machines. Of course, this mere theoretical possibility does not imply that it would actually occur; our authors do not provide any more evidence for this possibility than they do for the possibility of mind transfer being available to the general public. And our authors do intend the extraordinary future to encompass the possibility of widespread mind transfer, not just mind transfer for a lucky few.

10.7. Reliance on Future Technologies

Of course, the authors depicting the extraordinary future believe that new technologies will come to the rescue of Moore's Law. They cannot spell out exactly how this will happen in any detail, however, because technologies such as nanotechnology are still in their infancy. The willingness of the authors to believe in the power of future technology to come to the rescue, even without much evidence how or when it would do so, appears based in their belief that Moore's Law or the wider trend of which it is a part is some inexorable force. So instead of nanotechnology or the other new technologies cited being an established field with a proven track record, which can then be used to buttress the shaky claim that Moore's Law will continue indefinitely, it seems the belief in the inexorability of Moore's Law. On the other hand, the wider trend of which it is the latest manifestation) is used to buttress the claim that nanotechnology will produce what its proponents say it will. We have a lot of shaky arguments trying to support one another. Such faith reminds one more of religious fervor than of scientific impartiality.

Just what are these technologies? We have seen several mentioned. With respect to the route to increased computing power, nanotechnology is mentioned, as well as atomic computing, which might be considered to be part of nanotechnology. Nanotechnology concerns the technology and production of machines on the molecular or nanometer scale, though atomic computing involving quantum physics would seem to deal with what is even smaller still. Other buzzwords thrown around when convenient to be, tossed into the equation include artificial intelligence, artificial life, and the transmutation of elements. While our authors realize that each of these technologies individually have limits, somehow the combination of them all will allow each to complement one another in ways that will overcome limitations. Interestingly, the one “technology” here that has shown any concrete results at all, artificial intelligence, faces significant problems that are scarcely mentioned or instead glossed over by our authors.

At this point, nanotechnology is more of a research program than an established engineering field. We are nowhere near producing any little self-replicating machines. A recent critical article (Stix, 1996) in Scientific American brings up some of the relevant problems with nanotechnology. Manufacturing at the nano level would require treating individual atoms and molecules as construction elements in a tiny erector set. It is true some preliminary efforts along these lines have succeeded. For example, researchers manipulated 35 xenon atoms to form the letters “IBM.” They did this with a scanning tunneling microscope, which dragged the xenon atoms across a nickel surface. Nevertheless, this is a long way from creating a self-replicating nanoassembler. Proponents usually fail to mention that the xenon experiment was, done in a high vacuum at a super-cooled temperature, but both of these conditions are impractical for everyday nanomanufacturing of the kind envisioned in the extraordinary future. Atoms that can be, manipulated in this special environment are much too reactive with ambient atmospheric and other environmental elements to allow such manipulation to occur in normal environments. Other scientists accuse nanoenthusiasts of failing to provide crucial details about basic engineering that would need to be, done in nano manufacturing. For example, how close are we to realizing one of the exciting aspects predicted of nanotechnology, the ability of nanomachines (“nanoassemblers”) to reproduce by self-replication? At least one prominent chemist charges

that this is still sheer science fiction (Stix, 1996) [22]. There does seem to just, be a blind faith that we will be able to manipulate all kinds of particles at will to form what we want. However, chemical compounds follow the rules of chemistry and behave in well, known ways, including potentially reacting with everything around them, and it is not as simple as just grabbing parts of molecules and building whatever you want.

The *Scientific American* article of course generated critical responses from nanoenthusiasts. One can follow the discussion on the Web pages of The Foresight Institute, founded by K. Eric Drexler, a prominent proponent of nanotechnology.

We do not know that nanotechnology (or any of these other technologies) will not work; in fact, I hope it does. However, how much evidence is, provided by hope? While undoubtedly, development of such new technologies will proceed and they may in fact allow the production of smaller machines of various sorts, the hodgepodge of terms thrown around in this context seems more like the result of fervor or desperation than insight. None of our authors nor anyone else really knows what different form computers will take, if any, after silicon materials have played their last card. One reads encouraging reports of experiments involving some of these technologies, but we appear to be nowhere near ready to announce the arrival of any serious alternative kind of computer to the ones we have now. Just because scientists in the lab can get a few atoms to line up to spell “IBM” does not mean nanoassemblers and atomic computers are just around the corner. However, 2020 really is sort of just around the corner.

The authors in question rely on nanotechnology not just for new generations of computers, but also for the robot bodies that such computers will inhabit. Again, details are lacking because no one yet knows how to manufacture truly functional humanlike flesh (skin, muscle, nerve, and brain tissues, etc.) out of chemicals. If we were anywhere close to this development severe burn victims would not be forced to die while medical teams tried to cover large portions of their bodies with skin grafts and bandages. I read a news item the other day that reported an enterprising physician was using the knee joints of Barbie-dolls as artificial finger joints for people who had lost fingers and hands. A sympathetic Mattel toy company had at the doctor's request, sent him a box full of Barbie legs. Barbie legs? Mattel? Where are all the artificial organs and other replacement parts that we were supposed to have perfected by now? Human hearts and baboon livers seem to play more of a role in transplants than the synthetic models (artificial hearts are, used as temporary stopgap measures only). We seem pretty, far off from the imminent arrival of a practical macrotechnology for human parts, much less nanotechnology (remember that according to our authors, super smart computers may arrive as early as 2020). Even the ultimate nanoenthusiast K. Eric Drexler refuses to predict exactly when nanotechnology will be able to fulfill all of the predictions for it. Perhaps some important and encouraging research has been, done in these areas, but we are too far away to confidently, predict when we will be able to make a humanlike body, much less something better.

Nanotechnology advocates often cite a lecture that Richard Feynman gave in 1959 pointing to the possibilities of something akin to nanotechnology. In that lecture, Feynman allowed that he saw nothing in physical law that precluded making computers enormously smaller than they were then. While one wonders if what he had in mind was partially fulfilled in the silicon advances in microchips of the following decades, he does leave open the possibility of manipulation at the molecular and atomic level (Feynman, 1959) [23]. Nevertheless, the success of nanotechnology requires a lot more than just the pronouncement

of an eminent scientist that it is not physically impossible! Just because Feynman declared that such technology is not physically impossible, does not in it give us concrete reasons to think that it will be available anytime soon or even at all?

We do not want to sound too negative here. The fact that nanotechnology currently resembles a religion or cult as much as it does a scientific endeavor does not mean that it will not produce some amazing breakthroughs. I am certainly not interested in mounting an ad hominem attack against Drexler. However, we need to be realistic about nanotechnology prospects before relying on it or any other technology as the “magic silver bullet” that will jump in right when we need it to allow us to make super smart robots with synthetic bodies.

I will not continue my description of the debate on nanotechnology and its future beyond pointing out that nanotechnology and related technologies are still largely theoretical and speculative. The authors depicting the extraordinary future might be right that nanotechnology will rescue computing from its current silicon limits and provide robot bodies, but right now, those claims are as much science speculation as is the depiction of the extraordinary future itself.

10.8. Computing Power Alone May Not Be the Answer

While our authors realize that the brain appears to operate via something like a massive parallelism rather than by sequential processing, they do not seem to fully, appreciate that this may make the need for faster computers less relevant than the proper style of processing. Their preferred view seems to be that as long as the overall processing power of a robot brain is the same as that of the human brain, the robot will be as smart and capable as the human brain, even if the robot brain operates with less parallelism than does the human brain. However, the fact that this assumption may not be correct could have significant implications. This issue I will explore in this section.

Largely the authors' discussions of human brain size and anatomy seem to be in the ballpark. No one appears to have any basis for more precise estimates than a brain size of around 1011 neurons and thousands of connections per neuron. Perhaps surprisingly it does not seem to me that their estimates have to be very accurate. If computing power is advancing exponentially as fast as they claim, then even if they underestimate the brain's computing power by a factor of ten, say, that means only a few more years of development time to get a computer to that power. Their estimates vary among them by this much anyway. Of course, if they were wrong about the exponential growth of computing power, with it actually advancing in the future at an incrementally slow rate, then an accurate estimate of brain size would be more important. However, if the predicted rate is off by this much, their predictions will be off the mark by decades or centuries anyway.

The most significant problem with the viewpoint of our authors in this context, however, may be the use they make of their understanding of brain anatomy and physiology. The authors focus on estimating the number of neurons and connections in the brain in order to estimate the computing power of the brain. Discussions of the computing power of the brain emphasize the total number of brain events or neuron-to-neuron firings per second. Nevertheless, this focus may represent a failure to fully, appreciate the significance of the distinctive way the brain works. The authors are of course aware that the brain appears to

operate by a massive parallelism rather than sequential or serial processing. However, their talk about translating brain size into calculations per second or total storage in terms of bytes or bits seems to place more emphasis on getting an equivalent in raw computing power than in getting an equivalent in terms of massive parallelism. So my criticism here is that in spite of some of their comments, for the most part our authors seem to have their emphasis in the wrong place. In order to clarify what I mean, we need to discuss the difference between the traditional symbol, processing model of computing and more recently developed connectionist models.

Digital computers have become popular during the last decade or so as very many people use personal computers in their office or home. Of course they have been around longer than that, since around the forties and fifties, when they were too big and expensive for individuals to own. The earliest computers were, hardwired to run a particular program, but since then everyone has become familiar with the distinction between the program or software run by a computer and the hardware that runs the software. People are now familiar with the idea of being able to run different programs on the same computer, and some people are familiar with the idea of being able to run the same program on different computers. Which means loading the exact same program on two similar personal computers or running "equivalent" versions of a program on two different computers (such as an IBM-compatible PC and a Macintosh computer). In short, the distinction between software and hardware has become common knowledge. The idea that a computer could have a memory space that could hold different programs at different times was, developed by von Neumann and known as the "stored program" concept, such machines sometimes is being, called "von Neumann computers." A von Neumann machine is a sequential processor (carrying out operations in serial fashion); it stores data at particular memory locations, accesses this data via the addresses of such locations, and features the Central Processing Unit (CPU) as the single locus of control (Copeland, 1993, pp. 192-194) [24].

Now, the brain does not seem very much like a von Neumann machine for many reasons. First of all is the massive parallelism of the brain enabled by the great number of connections among neurons. As we have mentioned, a neuron may be connected to ten thousand others (or even a hundred thousand), and if the average is a thousand, and we have 1012 total neurons, we have 1015 total connections (Copeland, 1993, pp. 182-183) [24].

Memory storage also appears to be different. Modern computers separate the CPU, which does the processing, from primary storage (RAM). Some memory may remain on the CPU to speed up processing as a cache, but in any event, storage and processing occur in different places. It is not clear that the brain works like this, with memories seemingly distributed among synapses that may also be, used for processing. In other words, computer memory stores a datum at a specific address, whereas the human brain distributes a single memory over many sites (distributed storage) (Copeland, 1993, pp. 190-192) [24]. Our authors seem to realize this but this crucial difference with modern digital computers does not appear to unduly concern them.

There seems to be another disanalogy between computer and brain memory access. Computers store data to be accessible by address. The brain, in contrast, can access memory via content (content-addressable). Human recall of an event can be, initiated by any number of relevant triggers far from finding it by specifying a unique address where it is stored. Computers can retrieve data via something like content-addressable techniques, as in examining all address contents (way too inefficient) or even better in the use of hash-tables.

However, the disanalogy here is that the programmer must set up the hashing technique in advance, whereas human memory recalls events from any number of triggers, such as an associated smell or sound, various words or phrases, connections to other concrete and abstract ideas, etc. This open-endedness is not, captured by the use of a hashing technique (Copeland, 1993, pp. 188-189) [24].

Considerations such as these have led many in AI to look to parallel distributed processing rather than sequential processing as a model of how the brain works. The basic ideas are as follows. Parallel Distributed Processing (PDP) networks (connectionist networks) are built of a dense interconnected mass of simple switch-like units, artificial neurons as it were, that to use the simplest example, are each either on or off. A neuron fires if a sufficient number of neurons connected to it are themselves firing. Neural connections have different strengths (weights or conduction factors). The threshold of a neuron is the minimum input that will cause it to turn on. Some neural connections are excitatory and some are inhibitory. The networks switch themselves on and off in response to the stimulation they receive from their neighbors. When the total input meets or exceeds the threshold, it switches on, and when it drops below the threshold, it switches off. The patterns become complex, however, because the switching on of any single neuron may have an effect on a great number of others (Copeland, 1993, pp. 208-210) [24].

The network is, divided into an input layer, an output layer, and "hidden" layers in between. The input layers are set up so that they can be, switched on or off in a particular pattern irrespective of the influence of their neighbors. This pattern then is the input. The repercussions of this input pattern rebound throughout the network until stability is, reached, with some of the neurons permanently on and others permanently off. The output can then be, read off, of a bottom layer, which represents one edge of the stable pattern. Units can also be set up to act probabilistically to threshold values. Changing the strength of a connection can throw the network out of its current stability into more activity. The operator can produce a desired output from a given input by adjusting the strengths of the connections; one way to do this is, called "training." This is a systematic way of locating output units that ought to be different and adjusting the relevant connection strengths. Since each change usually induces other changes, this can be a lengthy process (Copeland, 1993, pp. 210-214) [24].

The above description fits the simplest example of a network, with neuron values either on or off. It is also possible to have the neurons select from a range of values between fully on and fully off. Meeting one of various thresholds may cause the unit to jump from one level to the next. The input and output patterns for a network composed of such units will be sequences of real numbers rather than strings of zeroes and ones (Copeland, 1993, p. 220) [24].

Therefore, PDP-connectionist networks are of interest because they seem closer to the brain than do von Neumann architectures. There are other key differences between a PDP-connectionist network and a traditional computer using the von Neumann machine architecture. The traditional computer operates by manipulating symbols, but the network consists of units exciting and inhibiting one another rather than program-governed manipulation of stored symbolic expressions. However, note that a simple network with on or off inputs can be regarded as a program-less bit-manipulator, with the input on and off states representing bit values. A coordinated array of networks can even simulate a von Neumann machine, with one network trained to perform assembly language level shifts, another to perform compare-rights, and so on (Copeland, 1993, 219-220) [24].

Now our claim is that while our authors concede that the brain uses a massive parallelism, closer to the PDP-connectionist model discussed above than to the sequential processing of a von Neumann machine. Their picture of the mind seems to see it more as the stored program type of software running on a modern digital computer of the von Neumann type. The distinction between software and hardware, and the idea that the software can be uploaded onto different computers, has become for some away (a model or metaphor) of thinking of the relation between the human mind and the human brain. Certainly, it has our authors in its grip. While they don't want to say that the mind is a "substance" distinct from the brain, and wish rather to think of the mind as in some sense nothing more than the brain or even identical to the brain, they also tend to think of it as the program that the brain runs when it is operating. On this analogy, the mind is to the brain as a program is to a computer and as software is to hardware. We need to keep this metaphor or model in mind in trying to understand the viewpoint of our authors. Our authors have humans in the future being able to "port" themselves all over the place from machine to machine, as the need arises. If they don't think of the mind as a piece of software, of the type of stored program that runs in a von Neumann architecture, it is hard to see how they could envision this sort of thing happening. But the mind as software is more than a metaphor to them. Their predominant view is that the mind is literally a program. For example, Moravec (1999, p. 210) [4] thinks that humans will exist in the extraordinary future as artificial intelligence programs running on platforms other than the human brain. As programs, our minds might be laser-beamed at the speed of light to inhabit distant robot brains (Moravec, 1988, p. 214) [25].

This equation of the mind with software might be the cause of an overly hasty identification of a brain neuron firing with a calculation. Recall that they attempt to determine the computational power of the brain in terms of total number of neuron firings per second, as if this should be merely, duplicated in a computer almost irrespective of the way the brain is structured or carries out its neuron firing. Looking at the brain itself, it is not clear in what sense a single neuron firing represents anything. But the tendency of our authors is to equate a single neuron firing with a digital computer carrying out a single calculation. We cannot find any argument even attempting to justify this identification. One can understand the natural tendency of trying to identify a neuron firing with something specific that can be, represented in a digital computer, whether it is a calculation, an instruction, a cycle, etc. But such identification really does seem to be just an assumption, and a questionable one at that. The representation or representations occurring when a computer carries out a calculation, whether just moving a bit into a register or something more complex, may or may not be what is occurring in a neuron firing. It certainly has not been, proved that a neuron firing corresponds to a representation of a proposition or the meaning of a word, and many think that any representation in the brain must be on a more global scale than a single neuron event.

Besides the view of mind as software, our authors may be in the grip of a view of the brain that sees it as a symbol processor, which fits in well with the former view since when a computer runs a software program it is manipulating symbols. To see how this view develops, consider how a computer works in general terms. The transistors of digital computers can be put in various electrical states of different voltage levels that can be thought of as representing "off" and "on," with off and on standing for the zero and one values of a binary digit. The computer works by carrying out operations (dictated by the application and operating systems programs) that ultimately shuffle these voltage levels around in a way that is, interpreted as carrying out operations on numbers. These numbers can be, thought of as representing other

characters, such as ordinary letters and decimal numbers (this is ASCII code). Therefore, for example, a computer spreadsheet appears to us to be able to add 2 to 1 and get 3, but what happens to accomplish this is that the program translates the decimal numbers and the plus sign ultimately into machine level binary number values and electrical states. It carries out the requisite processing by changing its electrical states in accordance with the instructions of the program, and then translates the relevant resulting electrical state back into a display of the sum. The shuffling of symbols in accordance with rules, carried out by the computer in its electrical state manipulation, comes to be, seen as computation.

An important assumption of what might be termed “traditional artificial intelligence” is the symbol-system hypothesis (SSH), which stems from this view of computers as engaged in shuffling symbols (Copeland, 1993, pp. 58-60). We can talk of a weak version of the Symbol-System Hypothesis (just abbreviated SSH) and a strong version (SSSH). The weak version holds that a symbol-system (such as a computer) can think, though there might be other things that think (Copeland, 1993, p. 82) [23]. While digital computers are engaged in a kind of thinking, other things that are not universal symbol manipulators might also carry out thinking. It may be that humans are such things, and so the fact that we can think does not prove we are computers or fundamentally just symbol-manipulators. In contrast, the strong version of the Symbol-System Hypothesis (SSSH) holds that computing is necessary for thinking, all thinking is computing, and only computers of some sort can think.

On SSSH, the computer becomes a model for the human brain, the computer software becomes a model for the human mind, and thinking is, seen as nothing but a type of computation analogous to what goes on in common digital computers. It's perhaps inevitable that this would happen; big old mainframes were referred to as “electronic brains” decades ago. However, “digital computer” here comes to refer to a class of objects that includes more than modern digital computers made of silicon and metal. Consider that computers might run in different ways that are equivalent in terms of producing the same output for a given input. This is the idea that you can run the same program on different computers. Going even further, it would even be possible to “run” the same “program” on something that is not an electronic computer as long as the relevant symbol manipulation were carried out. Therefore, we arrive at an abstract notion of a computer as a universal symbol manipulator. This allows humans to be, seen as computers. Since humans think, they must be doing so by computing in the above sense. Though humans do not work quite like modern digital computers do in the sense of manipulating voltage levels of silicon planar transistors, on this view our thinking does amount to some sort of manipulation of symbols. On SSSH then, not only is computation a metaphor for human thinking, humans (their brains or minds) are computers in this sense of universal symbol manipulator.

Leading advocates of SSSH who view mental processing as computation include Zenon Pylyshyn and Jerry Fodor. On this view, human thinking just is computation. The brain encodes its knowledge in the same fashion that a computer may be, said to use symbol-structures to encode semantic content (representations of things in the world). Human mental activity is the manipulation of sentence-like symbolic expressions that compose an internal mental language or “Mentalese” (Copeland, 1993, p. 181) [23]. Fodor holds that our brains have mental representations that are sort of like sentences or propositions. The sentences “It is raining” or “Il pleut” or “Es regnet” in particular languages can express the fact that it is raining; or rather, all of these sentences in particular languages express the same proposition that you believe. While the proposition is not literally in your brain or mind, a mental

representation corresponding to it is, and this mental representation can cause you to act in response, such as by putting up an umbrella (this theory is, called the "representational theory of mind"). These mental representations have a structure similar to language (the "language of thought" hypothesis). I manipulate these mental representations purely formally just as a computer manipulates symbols, and this manipulation is truth preserving (the so-called "computational theory of mind").

Now the question arises of whether our authors hold to SSSH or merely SSH. They don't explicitly use these phrases, but given their predominant view of the mind as a piece of software, it is quite natural to interpret them as holding to SSSH. While I don't want to accuse our authors of being Fodor disciples, if they do hold to SSSH this might be what causes them to neglect the particular way the human brain "wetware" works. The massive parallelism of the human brain might be seen as just the way our brains implement symbol manipulation; a robot brain could have the same mind and just implement the symbol manipulation is a different fashion. Our brains and the robot brains are both computers, but different computers can run equivalent programs, with in a sense the "same" program running on different computers, so they see no problem if the robot brain doesn't work in quite the same way the human brain does. As long as the symbols get manipulated in equivalent ways, thinking will occur. As long as the overall computational power is the same, other differences between hardware and wetware probably will not matter. Occasionally one does see the comment, as mentioned earlier from Paul and Cox, which parallelism may be important, and perhaps some robot brain parallelism will be, needed for computers to do what human brains do. But this may be more an acknowledgement that perhaps some form of parallelism is the best way to implement symbol manipulation, not that the essence of human thought might be radically different than symbol manipulation. Even then, we see no insistence that, for all we know, the parallelism of the robot brain might have to be as extensive as that of the human brain for it to be able to do what the human brain does.

Our authors would certainly hold at least to SSH, that computing is thinking, even if they might not unequivocally embrace SSSH if pressed about it. However, adopting SSH might not entail that just any symbol manipulation is thinking. It is not clear how much symbol manipulation must be, done for our authors to consider it thinking. The assumption seems to be that once the computer can match the processing speed and power of the human mind, then it will be thinking. Nevertheless, they do not seem ready to embrace every low-level type of calculating as an instance of thought. They are not necessarily all "panpsychists."

Copeland mentions three positions on the relation between the symbol-processing model and connectionism. Implementationalism holds that PDP is just how symbolic computation (manipulation) is realized in the human brain. So the brain, although it consists of PDP networks, is still a computer in the sense of a universal symbol system. This seems to be the position of our authors, though if so they do not seem to be aware that they are just implicitly assuming the truth of this position rather than arguing for it. They also do not seem to be aware that there are alternatives that argue that implementationalism is incorrect or even impossible. What is known as eliminativism (in this context; it is not exactly the same as eliminative materialism) holds that eventually we will see that the brain is not a symbol processor. We will be able to explain all forms of cognition without the use of any type of brain code, such as the Mentalese assumed by Fodor (explained above). This seems pretty far from what our authors are assuming, since they see such a parallel between the software that is running on digital computers and what is going on in our minds. A more eclectic position is

that of moderatism, which sees a variety of theories as necessary to explain the brain. Some processes such as language processing and logical reasoning will yield to the symbolic approach, while others such as face recognition and associative memory will be amenable only to pure PDP (Copeland, 1993, pp. 244-247) [24]. Our authors might consider being open to this latter position and to the weak form of SSH. Nevertheless, if so they would have to be careful. An assumption of our authors is clearly that for human-computer mind transfer to work the human mind, if it is not already just a piece of symbol-manipulating software, must be translatable somehow into software of some sort, so that it can be ported around to different platforms. Thus for our authors to be moderatists, the moderatist must allow that it make sense to talk of the software of the processes that are amenable only to pure PDP. If moderatism allows this, then they could be open to such a position. But it may be that the types of human brain processes that can't be captured by the symbolic approach are not readily thought of in terms of software at all, and these processes might then prove not amenable to transfer throughout the many machines humans as software would want to port themselves to.

It seems that on any position that seeks to combine symbol manipulation with parallelism, such as the implementationalism discussed above, inevitably there is going to be unclarity about how these two aspects work together, given our present state of ignorance of exactly what the mind and brain do in the activity of thinking. Though networks can be, used as symbol manipulators, they seem very different. However, the difference between PDP networks and symbol-manipulators in general is not clear-cut. This is because the nature of symbol-manipulators (as opposed to one kind called von Neumann machines) is open-ended. SSSH allows the brain's symbol-manipulating operations to be different in radical and presently unknowable ways from that of a von Neumann machine (Copeland, 1993, pp. 220-221) [24]. It is not, precluded that the brain may be using some kind of PDP architecture to manipulate symbols in some way. However, it could also be that the brain uses its PDP architecture with no symbols and no programs (Copeland, 1993, p. 221) [24]. Thus, the dispute between implementationalism and its opponents remains unresolved.

Our point in this section is not about whether our authors hold to SSSH or SSH, or even whether they are implementationalists or moderatists, but about how our authors may be in trouble by not taking seriously enough the fact that the human brain is so massively parallel. As I stated, our authors are aware of current theories of the brain that have it operating on a massively parallel scale rather than like a sequential processing computer, and they talk of the possible need to build a computer that uses parallel processing. On some mind transfer scenarios presented, by Moravec for instance, the transfer is conducted by building a robot brain isomorphic in some sense to the structure and functioning of the human brain (though it is not clear how fine-grained this isomorphism is supposed to be). However, they do often allow that less parallelism in the robot brain than in the human brain could likely be good enough. This is what leads one to think they are implementationalists at heart--the parallelism of the human brain is just the way the symbol manipulation of thinking occurs in humans, but in robot brains it could make do with less parallelism or perhaps none at all.

When we claim that it seems to me that they have failed to take the brain's massive parallelism seriously enough, by this I mean that they fail to appreciate that maybe the brain can do what it does just because it uses this type of processing, and so making do with less parallelism may not be an option at all. They under-emphasize the point that porting oneself around the universe into various platforms, some with less parallelism than the human brain

or no parallelism at all, may not be possible. This would explain their concern with Moore's Law continuing to hold well into the next century. Moore's Law, as it has evolved, is about increases in chip density that are relevant to processing power and in terms of the kind of computers that are being, developed to keep Moore's Law in force, this power is in terms of sequential speed. Strictly speaking, as Moore understood it, Moore's Law does not have anything to say about parallel processing within a chip, as long as the components on the chip make up an integrated circuit. However, Moore's Law as originally understood is certainly not about parallel processing among a large number of chips. It is about getting the most computing power out of a single chip. Thus, the concern is with moving to shorter paths via narrower circuit widths. Thus, the concern will be with continuing to pack more and more transistors on a chip. Of course our authors would reply that while they talk of Moore's Law, which is not about parallel processing among large number of chips, they think Moore's Law is just an instance of the wider phenomenon of exponential growth in computing power per se, which would cover parallel processing among large numbers of chips. However, as we have mentioned, the further one gets from Moore's Law, the harder it gets to provide evidence of any kind of real “law” at work moving us to more, powerful computers. It is difficult enough to make the case that Moore's Law is really a law.

Given the way computer chip technology is advancing, if Moore's Law does remain in force and powerful computers of the type our authors envisage do come about within the next half-century, it looks very likely that such computers will be sequential processors or at least make do with substantially less parallelism than that present in the human brain. But consider that a modern computer can already process serially many more times per second than a human brain neuron can fire. We already seem to have more than enough speed now to match what human wetware can do in terms of a neuron firing. Moore's Law could fail right now and we would have more than enough speed to match human wetware in this regard--in fact, we've had it for a long time. So why the need for still more processing power, in terms of more sequential speed, if we have already far surpassed neuron processing speed? Obviously because we have not yet matched the total processing power of the human brain, even though our current digital computers can far surpass the brain in terms of sheer serial speed. More processing power, in terms of still faster computers, would be useful only if one were trying to accomplish by some somewhat, sequential processing or minimal parallel processing what the human brain does by massive parallel processing. Because the human brain is so massively parallel, the total number of neurons firing per second is huge, even though individual neurons may not fire very many times per second. Instead of taking existing computer processing speed and recreating the human brain's massive parallelism, it would seem that at least much of the time our authors envisage trying to get the same overall speed by using much faster processors operating in a much less parallel fashion.

To take the brain's parallelism seriously would be to try to estimate what it would take to recreate it, and to estimate realistically as well, when this might be possible. It would mean not just assuming that we could probably translate the brain's parallelism into some form of sequential computing. It would mean placing less emphasis on Moore's Law as such, or an equivalent trend that focuses just on faster or more powerful processing per se, and instead placing more emphasis on advances in parallel computing. What needs to be, considered is not just Moore's Law but also any similar trend discernable with respect to increasing levels of parallelism in computing. Consider that the capabilities of the human brain and mind in terms of thinking abilities, and consciousness may arise from some physical properties of the

human brain related to what it is, made of, from the way that the brain is organized and functions in a massively parallel fashion. Alternatively, from something else, we have not thought of. We think it is safe to say that at this point we just do not know. The robot brain will not be, made of the same stuff as the human brain, so we may lose the possibility of such thinking and consciousness in robots if that is the relevant aspect that provides them in humans. However, suppose it is not and that fine-grained structural, organizational, and functional identity (in terms of parallelism) is instead what would provide for thought and consciousness in the robot. In such a case, if we do not have the same parallelism, we lose that kind of identity, and so then lose the possibility of the robot being able to think and be conscious like we are. As we discuss later, it may be that we cannot ensure that robots are conscious even if their brains duplicate the parallelism of the human brain on a very fine-grained level. However, to abandon such fine-grained parallelism really might make it even less likely, that the robot would be capable of our kind of thought and consciousness. So in the end trying to build the sequential processing robot equivalent of a fundamentally parallel processing human brain, even if that robot uses some parallelism, means not only that it may not work but also that we may have lost our only hope of providing for thought and consciousness.

The problem with the approach that focuses on obtaining just raw processing power in whatever form is that it neglects the possibility that perhaps humans are able to do what they do because of the specific way the brain works. The brain uses organic materials put together in a particular way and apparently operating in a massively parallel fashion. It does not seem to separate memory storage and processing components in the way a modern digital computer does. We do not know whether and how human capabilities and thought are, enabled by such factors. However, when designing a computer to match the intelligence and capabilities of humans, the further one departs from the actual design and materials of the brain, the greater one risks failure to match human brain functioning in all its varied aspects. To take the brain's parallelism seriously, our authors should consider that if we cannot match the way the brain works in a robot brain, we might not be able to provide the kind of robot we need. Since, as it explained more fully below, we do not seem to be anywhere near being able to match the brain's parallelism in a PDP computer, much less on a von Neumann sequential processor. Caution would seem to be more in order than their naive optimism that given enough of Moore's Law, our computers will be powerful enough to allow for the necessary robot mentality.

Taking the composite picture of the comments of our authors distilled into a common perspective, I think that this position of our authors is sometimes perplexing and perhaps inconsistent. Maybe this is a little unfair, since they don't always all speak with one voice, but I think all of them are more concerned to point to such trends as Moore's Law than they are to figure out whether we will really be able to crack the riddles of the brain's type of massive parallel processing. On the one hand, as we have seen above, they worry about getting greater overall processing power. But the way they see us getting more processing power is to extrapolate trends to pack in more chip density, etc. They don't claim that current chip technology should be overthrown and remodeled on the way the human brain works, rather, they seem to be concerned with continuing down that road until speed increases will allow the computer to match the brain. Thus, the concern is with Moore's Law and switching to atomic computing, when silicon etching reaches its limits. On the other hand, elsewhere, they sometimes talk of building a robot brain by “modeling” it on the human brain. In fact, as we

shall see in upcoming chapters, on some scenarios (but by no means all) the robot brain is pretty much an electronic copy, neuron by neuron, of the human brain. Here presumably massive parallelism would be a prominent feature of the robot brain. These two views seem hard to reconcile. It shows they really do not have a definite picture of how to pull off the project of a smart robot that will accept the transfer of a human mind. They have just the vague idea that it will be done, but they do not know how much parallelism will be involved, if any at all. But my advice is that if you really want to make a robot brain copy of the human brain that has a good chance of doing what the human brain does, forget Moore's Law and focus on figuring out how the brain's parallelism works and whether we have any realistic chance of advancing in that area with computer parallelism. We already have enough speed to match the brain--what we need is to figure out its organization and how, to duplicate it exactly. However, they have limited interest in this.

We fault our authors for being too optimistic that sheer computing power alone without sufficient parallelism will make for the smart robots that we need for mind transfer. But it's not as if they should assume all problems have been solved if they just insist that we use a massively parallel architecture. Apparently, we still have a long way to go in getting PDP-connectionist models to match the brain. Copeland (1993, pp. 221-225, 245) [24] summarizes the similarities and differences between the brain and current PDP networks. Similarities include the following:

1) Individual units in a network are somewhat analogous to neurons.
2) Human learning appears to involve modification of connection strengths in the brain.
3) Neurons behave in a roughly similar fashion to networks using input, excitation/inhibition, and output.
4) Both networks and the brain store information is a non-localized or distributed manner.
5) Human memory works through content-addressability, and networks can function this way (if part of a remembered pattern is given as input the completed pattern is generated as output).
6) Networks and the human brain "degrade gracefully."

However, there are important differences between current networks and the brain:

1) The brain features a diversity among neurons not matched in PDP networks.
2) Neurons are either excitatory or inhibitory, but PDP units have both functions.
3) It may be that the extensive repetition needed in an artificial network to train it up (give it the appropriate strengths) is not needed in human learning (it is at least debatable to what extent human learning involves such extensive repetition).
4) Network "learning" requires a trainer, specification of the desired output, and numerous adjustments after observing the wrong output; but nothing quite like this happens when a human learns.
5) The brain uses fifty different types of neurotransmitter (chemicals carrying signals), which is unmatched in a network.
6) The brain has an elaborate geometry of connections among neurons to near and far neighbors, which is unmatched in a network.

7) Humans are good at inference, and von Neumann machines are good at inference, but parallel networks are relatively bad at inference.

The suggestion I am making is that our authors not just mention parallel processing but take it seriously as the way to build a computer suitable for human-computer mind transfer. The relevant question then is not how far we can extend Moore's Law, seemingly to build a quasi-von Neumann type machine as well, but how soon we can hope to have the sort of massive parallelism in a computer similar to that operating in the brain. What they should be looking to find evidence for is some kind of “Moore's Law for Parallel Processing.”

10.9. THE MANY ASPECTS OF INTELLIGENCE

Recall that we are discussing the question of whether computers will soon be as smart as humans. In arguing that they will, our authors quickly launch into discussions of computing power advances and try to estimate the computing power of the brain. However, this is not necessarily the obvious place to start such a discussion. If the question is one of how soon computers will become as smart as humans, perhaps our authors should first or at least eventually consider what it is to be humanly smart rather than merely focusing on the recreation of the total processing speed of the brain in a computer. In trying to design a computer/robot for use in human-computer mind transfer, I would think it important to discuss the nature of the human intelligence you are trying to build the robot to have. I find this topic curiously neglected by our authors. Some of them do mention the question of what intelligence is, but it gets short shrift. For example, Kurzweil starts a section entitled “What Is Intelligence?” with the claim that intelligence is “the ability to use optimally limited resources” (Kurzweil, 1999, p. 73) [3]. Then he quickly changes the subject to one of how to solve intelligent problems (the answer is to combine simple methods with massive amounts of computation), as if intelligence is basically, the kind of problem solving a computer can do. Then, so much for considering the intelligence in humans!

We think the reason the subject of human intelligence is neglected is that our authors just assume that once you build a computer with the same overall processing power (or “overall” speed) as the brain, it will be as smart as a human. It's as if intelligence is simply how quick you think, so matching the overall speed will enable the computer to think as quick as a human. However, if this is their assumption, it is obviously flawed. With respect to overall computational speed in terms of total computations or firings per second (which they sometimes equate in a very questionable fashion as I have mentioned), our authors would estimate that the typical human brain is still far ahead of the ordinary desktop computer. Yet the ordinary desktop computer can do arithmetic of gigantic numbers far quicker than can the average human, and these examples could be extended to cover complicated calculations and feats of memory (and now even world-class chess if one considers Kasparov's defeat at the hands of an atypical computer). So how can the computer be smarter when it is slower (in terms of overall speed)? Obviously total processing speed in this sense does not track some types of intelligence. Also, consider that the brain achieves this massive overall speed by use of neurons that are individually quite slow, compared to electronic circuits. Furthermore, it should be obvious to anyone in computer science that hardware speed is only part of the

story. Two software programs running on the same platform can produce identical output for input and yet vary in speed.

Again, if you were trying to create a computer/robot to be as smart as a human did, it would not make sense to try to understand and characterize what human intelligence is, in all its varied facets. Anyone growing up in a public school system quickly learns that, some of the kids in the class are better in math than, others. That some of the kids are better readers, writers, or speakers than others, that some of the kids are better athletes, dancers, or actors than others, and that these superior groups do not always consist of the same kids. Intelligence may be more complicated than it first appears.

Our authors should realize that in their endeavor, human intelligence needs more attention than they give it because it is the proper measuring stick of relevant computer intelligence. This is for two reasons. First, the authors depicting an extraordinary future pick the intelligence of humans as the basis of comparison when they say computers will get really, smart in the next century. In spite of the fact the computers can beat human performance in some respects--calculating, as in the above example, and perhaps even chess nowadays--there seems the implicit assumption that overall humans are still the smartest being we know about. The second reason human intelligence is the measuring stick for computer intelligence in our context is that, it is humans, who are going to be doing the mind transfer into computers, and so we want the recipient of the transfer to come out of it at least as smart as when he or she went in.

It should be clear that we think our authors have neglected adequate treatment of the subject of intelligence. They might claim in defense that though this is true, the neglect has not harmed their analysis. I wish to argue that it has hurt their analysis by misleading them about how easy it would be to make a robot as smart as a human did. To show this I need to show that human intelligence might be more than just calculating and solving problems. I'll talk about a narrow understanding of intelligence, a broad understanding of intelligence, and some problems in getting a machine to achieve human intelligence.

The narrow understanding of intelligence is what many people commonly assume intelligence to be. It seems to be the position of our authors. On a common sense level many people would say that intelligence, or how smart you are, involves reasoning ability, calculating ability, the ability to understand, and so forth. We have what are commonly, perceived of as intelligence tests given to us while in school or when preparing to enter college or graduate school. The Stanford-Binet Intelligence Scale and the Wechsler Intelligence Scales have been available for years, and the Scholastic Aptitude Test and Graduate Record Exam incorporate many features found in IQ tests (Yam, 1998, p. 7) [26]. So since we have an intuitive understanding of intelligence, and we have tests to measure it, we might surmise there should be no real problem coming up with a definition of intelligence and a way to test how much of it robots have.

It is not quite as simple as this. There is no universal consensus on what intelligence is, and if we can't agree on what it is, we may not agree on what tests measure it. It may seem hard to believe that the notion of intelligence and its tests can be so controversial, but consider what your view is of some aspects of Sir Francis Galton's test for intelligence, administered between 1884 and 1890. What did Galton use? One test used a whistle to ascertain the highest pitch a person could perceive. He also had people pick up cases of gun cartridges (with the cartridges filled with either, shot, wool, or wadding) to see how well they could sort the cases by weight. Another test involved the factor of one's sensitivity to the

fragrance of roses (Sternberg, 1998, p. 12) [27]. To us these tests probably seem of dubious relevance, but to Galton these tests were good measures of intelligence. We can't say this is because Galton was just stupid; a 1926 estimate of Galton's IQ put it at 200, and even if the method used of estimating his IQ is questionable, he probably was no slouch. This difference of opinion about the relevance of these sorts of tests seems to reflect the fact that we don't all agree on what intelligence is or how to test for it.

Definitions and descriptions of intelligence in the AI literature vary. Sometimes the author of a paper just provides his or her favorite definition, with little or no justification. At other times the approach is more one of “here are several definitions, pick the one you like.” A recent AI work by Kelly notes the diverse abilities falling under the rubric of intelligence: ability to reason, use heuristics, do and know what is being done and why. Make use of knowledge, accept and interpret information, select appropriate information and apply it to problem-solving tasks, recognize and learn from mistakes, deal with unexpected or unusual situations. Learn and adapt to different circumstances, understand, reason, perceive, have insight, be aware of relevance, form adaptive responses, perform tests or tasks involving the grasping of relationships, meet novel situations, carry on abstract thinking, etc. (Kelly, 1993, pp. 38-41) [28]. Another approach, is noting that the concept of intelligence may not be one thing for all creatures. It prefers to think of intelligence as present whenever an organism has a quorum of the following attributes: verbal fluency, verbal comprehension, spatial visualization, perceptual speed, memory, reasoning, sensorimotor intelligence, symbolic thought, concrete operational thought, formal operations, and knowledge how. The knowledge that, ability to generalize, ability to learn from the past, ability to act purposefully, creativity, and the ability to notice significant facts (Kelly, 1993, p. 67) [28].

All these approaches above are attempts to define what I think of as a narrow understanding of intelligence. A narrow understanding sees intelligence as a single kind of thing and attempts to define it. As we have seen, definitions vary, but most of them stress things like abilities in reasoning, learning, or adapting.

Besides the above definitions, and the many definitions given over the years in psychology and education textbooks and the like, psychologists have at least several times surveyed experts specifically on the proper definition of intelligence. A famous symposium in 1921 in the Journal of Educational Psychology produced, fourteen definitions, many of which emphasized learning and adaptation rather than abstract reasoning alone. The range or flexibility of association, the ability to learn to adjust oneself to the environment, the ability to adapt oneself adequately to relatively new situations in life, the capacity for knowledge, and the capacity to learn or profit by experience. A 1986 update on the 1921 symposium solicited essays by experts in the field of intelligence including the invitation to again, try to define its nature. The understandings of intelligence discussed in this update were more diverse still than those of the earlier symposium, but some common themes hold between the two. Adaptation to the environment, basic mental processes, and higher order thinking (such as reasoning, problem solving, and decision making) were present in both. But important new emphases in the update include meta-cognition (knowledge about and control of cognition), the role of knowledge and the interaction between knowledge and mental processes, and the role of cultural context (Sternberg, 1990, pp. 35-36, 49-53) [29].

Unfortunately, the discussion above may be naive in simply assuming that with intelligence we are dealing with the individual. While nonspecialists may assume intelligence characterizes an individual, wider emphases from the updated symposium are instructive of

the diversity of alternative understandings. Three main “loci” of intelligence were involved: intelligence within the individual, intelligence with the environment, and intelligence in the interaction between individual and environment. However, numerous distinctions were made even further within each of these areas. Individual intelligence was discussed on the biological level, the molar level (the molecular level in the sense of the parts or components making it up), and the behavioral level. Biologically, comparisons can be involved within or among species or generations of species, and involving interaction with the environment. Within organisms, the role of structural and evolutionary aspects of the brain or process aspects of the neurons may be, discussed. At the molar level, the emphasis may be on cognitive or motivational factors. Cognitive processes mentioned include selective attention, learning, reasoning, problem solving, and decision making. Motivation theorists argue that motivation is involved in intelligence as much as cognition. Motivation to cognize may be relevant to the quantity and quality of cognition. As one might guess, the behavioral level of analysis focuses on what a person does rather than on what is being, thought. A major controversy is about the breadth of the domain of intelligence, for example is artistic or dancing behavior part of intelligence or within another domain? Furthermore, theorists do not agree on the relevance of everyday tasks to the issue of intelligence, some saying it is irrelevant to anything important and others arguing that a true understanding of intelligence is, found in such mundane behavior. Some theorists view intelligence as residing not in the individual but in the environment as a function of one's culture and society or of one's niche within these. Some would argue that intelligence is wholly, relativistic with respect to culture, and so it is impossible to understand intelligence without understanding the culture. Through its labeling and attributional processes, the culture determines the nature of intelligence and who has how much of it. What the culture, society, or niche deems intelligent will generally be a function of the demands of the environment in which the people live, the values held by the people, and the interaction between demands and values. Societal values that are in demand but not easily filled may come to be valued highly. Other theorists would claim that intelligence resides not wholly within the individual or environment but within the interaction between the two. A person may be differentially intelligent within different environments (Sternberg & Detterman, 1986, pp. 3-9) [30].

The above discussion presents a glimpse of a larger understanding of intelligence, one that sees it as more than just one thing. However, just to show you that this narrow understanding is still popular, let me highlight a recent article from a special issue of Scientific American on intelligence. This article, by Linda Gottfredson [32], claims that, there is a general intelligence (“ g “), that is depicted by a person's Intelligence Quotient (IQ). Gottfredson claims there is clearly a general mental ability, a “global factor that permeates all aspects of cognition,” that is measured by IQ tests and acts to predict success and performance in life. It was long ago recognized that in mental tests designed to measure particular domains of cognition, such as verbal fluency, mathematical skill, spatial visualization, or memory, people doing well on one test tend to do so on the others, and similarly for those doing poorly. The overlap or inter-correlation suggests that the tests are measuring some global element of intellectual ability. This general factor, or g , isolated by statistical factor analysis, is now, used as the working definition of intelligence by most intelligence experts. Particular tests also measure specific abilities, but these impurities can be, statistically separated from g . The g factor is especially important in behaviors such as

reasoning, problem solving, abstract thinking, and quick learning. While the concept of intelligence, and how people in a society are, ranked according to it could be "social artifacts," the fact that g is not specific to any particular domain of knowledge or mental skill suggests that g is independent of cultural content including beliefs about intelligence (Gottfredson, 1998, pp. 24-27) [31].

We turn now from the narrow type of view to consider the broad type of view. Most of the above theories saw intelligence as just one thing; they just differed on what that thing is. The broad view sees intelligence as not just one thing. Howard Gardner, for instance, thinks that human intelligence encompasses a set of competencies far wider than those captured in the notion of g .

Gardner holds that intelligence consists of a wider and more universal set of competencies. He refers to them as multiple intelligences. He developed his view after working with both gifted children and adults who had suffered strokes that shut down particular capacities while leaving others untouched. Each of the people he saw from either group had various strengths and weaknesses, and he noticed that a strength or weakness could exist simultaneously with varied sets of abilities and disabilities in the same individual. Gardner began to think that humans are more accurately, thought of as possessing a number of relatively independent faculties instead of having an "intellectual horsepower" that can be channeled in one direction or other (Gardner, 1998, p. 20) [32].

Gardner thinks his view fits in well with developments in various related sciences. Neuroscience knows of the modular nature of the brain, and evolutionary psychology holds that different capacities have evolved in particular environments for specific purposes (Gardner, 1998, p. 21) [31].

Gardner also makes two strong claims about multiple intelligences. First, all humans possess all of them, but second, not everyone has them in the same proportions; we all have different profiles (Gardner, 1998, p. 21) [32].

Of course, one could try to reduce Gardner's many intelligences to one thing, as something in common or behind all the different intelligences. Gardner himself says intelligence is "a psychobiological potential to solve problems or to fashion products that are valued in at least one cultural context." However, this misses the thrust of his approach. How many, intelligences are there? In considering whether a candidate capacity is intelligence, Gardner drew upon work in psychology, case studies of learners, anthropology, cultural studies, and the biological sciences, and he chose as criteria eight different factors. For a candidate capacity to be, considered a type of intelligence, it would have to possess many of these factors. These factors are potential isolation by brain damage. The existence of prodigies and savants with the capacity is an identifiable core operation or set, such as a musician's sensitivity to melody and rhythm. A distinctive developmental history within an individual of the capacity and a definable nature of expert performance of it, an evolutionary history and plausibility, support from tests in experimental psychology and from psychometric findings, and susceptibility to encoding in a symbol system. He finds there to be eight distinct and independent forms of intelligence or such ability: linguistic, logical-mathematical, spatial, bodily-kinesthetic, musical, interpersonal, intrapersonal, and naturalist. Existential intelligence is under current consideration (Gardner, 1998, pp. 20-21) [32].

Linguistic intelligence concerns the ability to acquire, form, and process language. Included here are abilities in symbolic reasoning, reading, and writing (Wilson, 1998a) [33].

High intelligence in this domain means a "mastery and love of language and words with a desire to explore them" (Gardner, 1998, p. 22) [32]. Logical/mathematical intelligence concerns the ability to think logically (especially inductively and to some degree deductively, which seems to involve linguistic intelligence), to recognize patterns (geometric and numerical), and the ability to work with abstract concepts (Wilson, 1998a). It also involves discerning the relations and underlying principles behind objects and abstractions (Gardner, 1998, p. 22) [32]. Spatial intelligence concerns the ability to perceive images, recall visually, and imagine visually (Wilson, 1998a) [33]. Musical intelligence concerns the ability to create and interpret music and discern differences in speech patterns and accents (Wilson, 1998a). Thus listening is involved, not only composing and performing (Gardner, 1998, p. 22) [32]. Bodily/kinesthetic intelligence concerns the ability to control fine and large muscle movement, and to create and interpret gestures and communicate through body language (Wilson, 1998a) [33]. Interpersonal intelligence concerns the ability to understand and communicate with others and facilitate group processes. Intrapersonal intelligence concerns the ability to have a strong sense of self, leadership abilities, intuitive feelings, a feeling of inner wisdom, and precognition. (Wilson, 1998a). Naturalist intelligence, recently added as an eighth type of intelligence alongside the original seven, concerns the ability to cope with environment in the sense of identifying and classifying natural patterns, such as those in flora, fauna, and weather patterns (Wilson, 1998b). Existential intelligence, under consideration, captures the human tendency to raise and ponder fundamental questions about existence, life, death, and finitude (Gardner, 1998, p. 21) [32].

Gardner's work is controversial. Some would consider his intelligences merely different talents. To Gardner, this devalues musical or body-kinesthetic abilities by implying that orchestra conductors and dancers are "talented but not smart." "In my view, it would be all right to call those abilities talents, so long as logical reasoning and linguistic facility are then also termed talents" (Gardner, 1998, p. 21). Gardner is recommending a wide notion of intelligence, one that incorporates a "range of human computational capacities, including those that deal with music, other persons and skill in deciphering the natural world." However, he still wishes it not be, conflated with other virtues such as creativity, wisdom, or morality (Gardner, 1998, pp. 21, 23) [32].

If the above long discussion of intelligence proves nothing else it shows that we should not take for granted that there is a consensus on the meaning of the term "intelligence." As mentioned several standard tests have been developed that test for IQ, but whether these are good tests for intelligence depend on one's understanding of intelligence. If you think intelligence involves g, then IQ tests will be good tests for intelligence. If, like Gardner, you think that intelligence is far more diverse, and then IQ tests will not be good tests for all the varied kinds of intelligence. If, with some of the approaches mentioned above, you think that intelligence involves the environment and society as much as the individual, then you will probably think the standard types of IQ test do not test for the right kinds of traits abilities. It may be culturally, biased in favor of a particular society's notion of intelligence, and they may neglect the importance of the role of motivational factors.

Can our authors get away with insufficiently considering the subject of intelligence? I tend to think not, but we are not entirely sure. It depends on what is going to happen to bring about the extraordinary future. Part of the extraordinary future involves creating smart robots. If this is to come about through traditional artificial intelligence means, then it would seem relevant to know what human intelligence is and how to test for it. Otherwise, how do you

know what to build into the robot? As we saw above, traditional artificial intelligence has thought it necessary to grapple with the notion of intelligence at least to some extent, whether or not they have advanced the issue.

On the other hand, some depictions of the extraordinary future see us creating smart robots on the fly as we transfer our minds into them. The way this happens is that an electronic (or whatever) copy is made of the human brain as the brain is scanned. We will see in a later chapter that the subject of exactly how this is supposed to occur is complicated, but on some versions, we don't really have to solve the problems faced by traditional artificial intelligence efforts in creating smart computers. We may not even have to know what human intelligence is. All we have to do is to translate the programs running in the human brain into their equivalent in a robot brain. (This assumes there are such programs.) Then again, in the testing of this equivalence we might be faced with questions of what exactly we need to test, and then the question of intelligence might rear its head after all.

In one respect, we certainly need to get a handle on what intelligence is and how to test for it. Our authors frequently claim that robots will not only equal human intelligence but will far surpass it. I do not know how you could show this, or even provide the statement with a concrete meaning if challenged, without coming up with some characterization of intelligence, and how to test for it. Giving the robot one of the standard IQ tests obviously presupposes that such a test tests for intelligence. However, could you really give a robot a standard IQ test and derive meaningful results? Standard IQ tests are of course very anthropocentric, expecting you to have grown up in a world of human culture, understand human language, etc. So they don't rate you on any absolute scale of intelligence that would, for example, let you know how much smarter you are than non-human creatures. In other words, we might know that I am smarter than the next guy, for instance, but I don't know how much smarter I am than a particular chimpanzee, or how much dumber or smarter I am than my PC. Likewise the standard IQ test might not let me know how much smarter than I the computer is. Even Gardner's wide notion of intelligence is, meant to characterize human intelligence--he is not considering the intelligence of other types of beings. In the above discussion of diverse approaches to intelligence, we observed that some psychologists were interested in comparisons of intelligence among species, but there is probably not some clear-cut test available, whose results allow us to compare diverse creatures. We could specify an ad hoc indicator such as the ability to use language, but this would probably be, seen as presupposing a particular narrow conception of intelligence that might be, biased against particular species. It does not seem that there has been much demand for the development of such a cross-species test, though I guess someone or other working in the field of animal intelligence has tried to devise one. So while it is or should have been the intent of IQ tests to refrain from cultural, ethnic, or racial bias, one can hardly fault the authors of such tests for leaving them biased species-wise. Computer keyboards might be a real problem for other species to master too, but there hasn't been much demand for typing parrots, for example, so one can't really fault keyboard designers for biasing their products against birds. However, if we get the robots of the extraordinary future, won't we want to try to work on this issue?

There is still a further reason for our authors to pay attention to intelligence. Human-computer mind transfer is, seen as a way to personal immortality, but it is also, seen as a way to make one smarter. If we are transferring our mind into a new brain, why not make it a better brain. One that makes us smarter? Our authors discuss this at great length as their imaginations run wild. After the transfer, you can give your brain all of human knowledge,

and spend your time calculating things that were previously way beyond your abilities. We will see such talk in later chapters. Obviously the improvements we would wish to make to the robot brain, visa a versa the human brain it replaces, would have to do with making the person transferring in to be smarter. However, how could we know in what ways to change the robot brain to make it smarter without knowing what intelligence is or how to test for it?

So where are we at this point? I have discussed the many definitions of intelligence above, especially Gardner's, to emphasize the varied types of intelligence (or the varied talents and capabilities in which one uses intelligence) humans possess. Certainly, the assumption should be that any robot we build for human-computer mind transfer be capable of exercising these kinds of intelligence. And any robot presented for inspection as a candidate should be able to pass tests to show this, if such a thing is possible. Certainly if we make the claim that computers will be as smart as humans will in 2020, for instance, we should back this up with an explanation of what we mean by “intelligence” and how we will know when computers have it. This goes as well for any claims that robots will be smarter than humans will or that we will be able to arrange it that humans after the transfer will be smarter than they were before.

10.10. THE TURNING TEST

The gist of the above discussion is that it is probably important for our authors to arrive at the proper characterization of intelligence. However, some Artificial Intelligence (AI) theorists believe it is not important at all. Haugeland asks, “How shall we define intelligence? Doesn't everything turn on this?” and then answers “Surprisingly, perhaps, very little seems to turn on it.” This is because “for practical purposes” we already have the entire test we need in the Turing test, a criterion that “satisfies nearly everyone” (Haugeland, 1986, p. 6) [34]. We think Haugeland is being too optimistic here. To claim that the Turing test is a good practical test of intelligence clearly presupposes a particular view of intelligence that may not be correct, whether for the purposes of our authors or artificial intelligence efforts in general. For instance, it may not be a very good practical test for Gardner's musical intelligence.

In some cases, we can imagine how to test computers or robots against humans to determine whether one has more of something. There could be a calculating contest, for example, that might easily show who is best. However, in other cases, it's not as clear, particularly if the human and robot do not have the same type of bodies. For example, there are certain industrial machines, whether forklifts, robotic arms, or cranes that are clearly stronger than humans are, when it comes to lifting heavy objects. They can lift thousands of pounds. The most a human can lift varies with the type of lift. Consider the type of lift known as a “dead lift,” where you just stand up from a crouch while picking up a heavy barbell from the ground without trying to lift it over your head or even to your chest. The most the strongest human can manage is going to be about 900 pounds or so, and the average human will struggle with a few hundred pounds. However, while a machine is often clearly stronger than a human is, you might have trouble coming up with a suitable test to demonstrate this. How exactly does a forklift do a dead lift when it cannot even crouch? While we are not interested in strength tests per se, but rather intelligence, we might run into similar problems devising tests for showing that a computer has all of the various types of human intelligence if

the computer does not have a very human-like robot body. Consider Gardner's notion of bodily/kinesthetic intelligence as the ability to control fine and large muscle movement, and to create and interpret gestures and communicate through body language. Can a computer without a human-like body have this type of intelligence? If it could, how do you demonstrate this. No one considers a PC especially talented at providing an interpretive dance, for instance. Given the apparent multiple types or aspects to intelligence, it is certainly difficult to imagine one test that would do the trick. Gardner himself is very skeptical of the ability of any standard type of intelligence test to accurately measure intelligence.

Alan Turing imagined a test that might help us out. This test has of course come to be, called the Turing test, so we should consider whether it fits our needs. (We could use it to determine whether a computer is intelligent enough for us to use it for mind-transfer, for instance, and then use the test again on the post-transfer being to see whether it really is still intelligent and thinks.) What has come to be, called the "Turing test" first appeared in a very famous paper by Alan Turing in which he discussed the question of whether machines could think. That particular discussion of whether machines can think is very short--Turing spends one paragraph on this question before replacing it with another. The question of whether machines can think apparently had terms too vague or ambiguous (such as "think" and even "machine") for Turing's patience. Perhaps it is just as well, because it is not clear to me what Turing had in mind by the term "think." He might have meant "think" in the sense of having consciousness. We wish he had considered the original question, which we think may be important to answer if "think" involves consciousness. In any event, later in the article he declares the question too meaningless for discussion. The new question he replaces it with has to do with whether a machine could fool humans in a variation on a certain sort of "imitation game" that was popular at the time.

The imitation game Turing describes is, played with three people: a man, a woman, and an interrogator. The object is for the interrogator, who is in a separate room from the man and woman, to determine by the end of the game which is the man and which is the woman by asking questions of them (via teletype or some other means that conceals their voices). One player is supposed to try to help the interrogator guess correctly (in Turing's example, the woman) and the other is supposed to try to help the interrogator guess incorrectly. Turing thinks that the player trying to trick the interrogator will want to lie, if that helps, but the other one will likely tell the truth (Turing, 1950, pp. 53-54) [35].

Now what Turing wonders is what will happen if a machine (computer, for our purposes) takes the place of one of the players (in Turing's example the man). The question "Can machines think?" is now replaced with "Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman?" From Turing's description, one wonders if perhaps the machine's job is to try to trick the interrogator into thinking the machine is male, rather than female, but other remarks suggest that the interrogator is rather trying to guess which one is the machine and which the human. In Turing's example, the man is trying to fool the interrogator into thinking he is a woman, and so when he is, replaced, we have the machine trying to fool the interrogator into thinking it is a human. The best strategy for the machine is probably to try to provide answers that a human would give; obviously showing amazing speed in calculations would catch the machine out pretty, quickly (Turing, 1950, pp. 54-55) [35].

How will the machine win the game? Turing's above comment suggests that it will win when it can match the frequency with which humans fool the interrogator about their sex. I do

not know what that frequency is. One would think that the interrogator would have a fifty percent chance of getting it right on guessing alone, so if the interrogator got it right sixty percent of the time, for example, the player would have done a pretty, good job of fooling him or her. Turing's opinion was that in about fifty years from when the article was written (it was published in 1950), computers would be able to play the imitation game so well that an average interrogator would not have more than a seventy percent chance of making the correct identification after five minutes of questioning. By that time, we will be attributing thought to machines (Turing, 1950, p. 57) [35]. We do not know how he came up with these numbers, but we imagine that with any questions fair game to be asked of the machine, if the machine can fool the interrogator the requisite 30% of the time it is doing a pretty good job of at least imitating thinking, if not downright thinking.

Turing considers a number of objections to his claim that we will appropriately use the term “thought” to refer to such machines as can win the imitation game. Many of them are the kind of objections to machine thinking that are sort of old hat nowadays.

The theological objection claims that machines cannot think because thinking is a function of the soul, and God gives souls only to humans. Turing is “unable to accept any part of this” but tries to reply to this camp on its own (theological) ground. Would not this restrict God's omnipotence? If a mutated elephant had a capable brain, would not God have the freedom to give the creature a soul were he wont to do so? Why not for a computer then? We would not be usurping God's power to create souls any more than we are when we take the appropriate steps to give rise to a child (Turing, 1950, pp. 57-58) [35].

The argument from consciousness claims that machines will not be able to match our power to be conscious, or at least that we will never know that they will, and so they cannot be said properly to think. Turing thinks that we will not need to answer the question of machine consciousness to decide the issue of whether a computer thinks, partly because the only way we could ever really know whether anyone is conscious would be to be that person, which we cannot do with other people (Turing, 1950, pp. 59-61) [35]. We agree that we will not have to know about computer consciousness to decide the question of computer thought, but we do not think it has anything to do with Turing's reason. Rather we can give some meaning to the notion of unconscious thought.

Lady Lovelace's objection amounts to the charge that a machine can never do anything original. Turing notes that we cannot be sure that originality on the part of humans is anything more than humans producing, unexpected outcomes from the application of earlier teachings or general principles (Turing, 1950, pp. 63-64) [35].

The argument from informality of behavior claims that there is a crucial difference between humans and machines with respect to following rules or laws of behavior. We cannot be machines because our behavior can never be, captured by a system of rules, which is what machines must follow. Turing thinks we cannot be sure there are no such rules yet undiscovered by science (Turing, 1950, pp. 65-66) [35]. Whether human mentality can be, captured in a set of rules seems to be a significant topic in artificial intelligence but we will not have time to go into it.

To our knowledge, no computer has yet passed the Turing test, though of course some computers have fooled some people, as in the case of ELIZA. Given this fact, two questions come to mind. First, will a computer ever pass it? The second question is “What would it show were a computer to pass it (what is it a test for and is it a good one for that)?”

However, before we get to these questions, we might pause to consider how seriously we should take Turing's claims. Turing seemed to think that a computer passing such a test would be, considered to have been thinking. This would mean that passing the test was a sufficient condition of possessing thought, or intelligence or whatever he considered the test to be a test. In addition, he predicted that this would happen by the turn of the century. Since no machine has really passed the test as he conceived it, and we still seem years away from having a machine that will, he was wildly off in this prediction. By all accounts, Turing was a brilliant fellow, but the utter failure of his prediction to come true makes him look like a little foolish. Maybe there is a lesson for our authors here.

Nevertheless, it is not clear how serious we should take the prediction. A friend of Turing who discussed the article with him claims that Turing thought of the article as basically propaganda (Gandy, 1996, p. 125) [36]. Turing was enthusiastic about the notion of machine intelligence and wanted to spread this enthusiasm. As well, it should be, noted that at the time Turing wrote the article there were only four computers in existence: in Britain the Manchester Mark I and Cambridge EDSAC, and in America the ENIAC and the BINAC (Copeland, 1993, p. 9). How accurate and even how serious could such a prediction have been, given the primitive state of computers when it was, made? Consider the seriousness of someone making an analogous prediction that by the end of the century a land vehicle would break the speed of sound by going over seven hundred miles per hour, if the prediction were made at the turn of the century when only a few cars were around and, then current record was sixty-five miles per hour. How could someone have enough evidence upon which to base such a prediction? (Incidentally, the speed records we refer to are, held by a jet-powered car, while the record at the turn of century and were, held by an electric car. Many records in between were held by cars powered by internal combustion engines. In 1900, who could have predicted jet engines?).

Just as we should not make a big deal about the inaccuracy of Turing's prediction, perhaps we should not worry too much about some details of the test. Just where? Turing did get those numbers of seventy percent, and five minutes? Why not sixty percent and ten minutes, which would be a little harder for the machine.

Note also that Turing did not say that if a machine passed the test this would show the machine could think like or as well as a human, only that we should say it could think. However, in the case of human-computer mind transfer and predictions about computers equaling and even surpassing human intelligence, we really are interested in whether the computer could think as well as a human. (As we argued above, we need to know something about intelligence to provide for this.) Therefore, we might wish to make the test even harder--the interrogator must guess correctly only fifty percent of the time in a test extending for an indefinitely long period--say, several days! We want a test such that passing it will show the machine can think (or possess intelligence) at least as well as a human, not just which it can think.

But the Turing test is widely discussed and so I will consider it to see if it is what we want. Our first question, then, is that of whether the Turing test will be, passed by a computer. Our authors think so, and think it will be, passed within a few decades. Though this thesis considers human-computer mind transfer and not computer intelligence per se, if the extraordinary future comes true when our authors say it will then the Turing Test will soon be, passed. On the other hand, if their claims that smart robots will be around soon are a bit optimistic, as I argue, then who knows when the Turing Test will be passed, if ever?

Why is it that no machine has yet passed the Turing test? There are several reasons. First, the computer would have to “speak” a natural language in order to converse. Getting a computer to understand and generate sentences in a natural language is an extremely difficult task. Just consider problems in understanding the meaning of a sentence. The computer has to take each word and determine the part of speech. Some words can be any of several different parts of speech, so the larger context must be, used to eliminate this ambiguity. The sentence must be, classified as a statement, question, exclamation, etc. All these steps must be, done to arrive at the syntax of the sentence. However, what does it mean? The computer must have some way of determining the semantic meaning of each word. Again, many words have numerous meanings, so context must be, used to disambiguate the sentence in this regard. The words must be put together to determine the semantic meaning of the sentence. However, this may require a consideration of the overall context--the place of the sentence in the larger conversation. In a typical conversation, we assume that the other conversant has background common-sense knowledge of the world, and so our comments are not always completely explicit. We often leave things unstated; assuming the listener will supply what is, needed to make sense of the remark.

We do the above process constantly in understanding the utterance of a sentence, but how do we get a computer to do it? We cannot just supply a dictionary and leave it at that, for this will not enable the computer to do the requisite disambiguation. We need to provide the relevant background knowledge. However, it is not clear how to do this. How do we organize all the facts about the world that we already know and present them to the computer? How does the computer determine which facts it needs for disambiguation? How does the computer find the facts it needs when it needs them in a split second (without searching through all of them)?

Assuming we could do all this, we have to provide the computer with some mechanism by which it can generate sentences to keep up its end of the conversation. It might have a lot of facts, but will it be able to respond to questions and comments the way a human does? Humans have firsthand experience of the world, their feelings, etc. Would this be in the database that the computer can access? Can and how a toothache feels be put in the database for a computer to access? Will it understand what the pain is like if it cannot feel (we are not saying it can or cannot)? Will it really know what words refer to if it has never had any firsthand experience of the world? Maybe what needs to happen is for the computer to become a subject in its own right, and train it about the world through some massive learning process to recreate what we all go through in childhood. Do we need to provide different kinds of sensory apparatus for it to do this? Even were we to get a computer to “speak” a natural language, it might not be able to pass the Turing test unless it could learn firsthand and gain experience of the world. We might think of all hurdles that need to be, overcome for a robot or computer to really, be as smart and able as a human is as the “classic” or traditional problems in AI.

The above considerations would seem to apply if we are to get robots smart enough to pass the Turing test. Our authors provide no insight on how to overcome these classic problems in artificial intelligence. They seem pretty, unconcerned about them. Nevertheless, we do have to distinguish between robots that get smart on their own and robots that are smart because their brains are copies of the human brain. It does seem that our authors believe that robots will be smart before we do any such copying, though it is still not clear how this will occur. But conceivably if somehow we could create a robot simply by copying a human, so it

might be assumed, we could do so without solving such problems. If a robot is, created by copying a human, then we do not have to solve the classic problems in AI, because they will have been, solved in whatever way the human brain has already solved them.

Here we see what seems to us, a tension within the depictions of the extraordinary future by our authors. Through all the varied discussions there seem to appear two distinct methods of getting a smart robot. First is the method of gradually solving the problems faced in AI research in the process of building better and better robots. The robots gradually get more capable until sometime between 2020 and 2050 they are as capable of humans, after which they get even smarter. The second method is just to create a robot by modeling the robot brain on that of a human. There are various scenarios for this that we will discuss in a later chapter. However, basically we start with a detailed knowledge of the human brain and how it works, reverse engineer the software running on it, and then write the equivalent robot brain software as we build equivalent robot brain parts. What we get on this view is somewhat an electronic (if that is what the robot brain medium winds up being) copy or replica that is equivalent in output, function, structure, algorithms, etc. to the human brain. It might be smarter because it will be running on better hardware (faster circuits, etc.). On the other hand, in the copying process, we might be able to improve performance in other ways, such as writing more algorithms that are efficient. Just as an aside, it is not clear that just replacing human wetware with some sort of electronic or atomic equivalent is going to make the robot smarter. We all have the experience of knowing people who seem smarter or dumber than we are, but we do not think this is due to such people having "circuits" that run faster or slower than do those of our brains. So likewise, faster circuits may or may not be what makes for more intelligence.

The tension between the two methods above for creating a smart robot is that we seem to need each method to be finished in order to get the other one working, which is impossible. Our authors do not really supply crucial details about how we are to do either, much less both. However, they do jump around from method to method, and among various options within each method, without saying, how exactly the details are to be pulled off.

Consider how each method is to work. On the first method, we solve the classic problems of AI. How is this, accomplished? Of course, our authors do not know, because if they knew, they would publish the results and build the robots. Paul and Cox mention the possibility of a bottom-up approach that recreates evolution, but details are not, provided. So instead, we get a depiction of gradual developments during the next several decades showing the arrival of computers and robots that are more and more capable, without any concrete details of how we are going to accomplish the amazing feats we will have to accomplish. Moore's Law, etc. says computing power will increase, and nanotechnology will come to the rescue of silicon limits, etc. so we will just be able to do it. However, we do not really have any reason to know that. What seem insuperable problems in AI are just going to be, solved that easily, do we? What we need, of course, is the super smart robots to already, be here to show us how to do it. Therefore, we really need the other (second) method already done to give us the robots to overcome these AI and materials problems of the first method.

On the other approach, we just copy a human brain and make a robot equivalent. This bypasses the need for solving the problems of AI, because in whatever way our brain does this, the robot will do this. Therefore, on this approach we do not have to slog though decades of AI breakthroughs. However, how are we going to pull off this second method? Where do we get that incredible knowledge of the human brain functioning and software design to reverse engineer the code and write the robot brain equivalent? It may be that extensive

knowledge of the brain alone, even assuming we had this, would not do it, because we need to write the software of the brain and then the robot equivalent. However, how could we possibly know what kinds of code does what kinds of things in the brain to enable humans to do certain things. Like access knowledge when we want it), and then know what kinds of code are going to do the equivalent in a robot brain made of different materials, without having worked through the solving of the classic problems of AI? I'm not sure we could. You can see a human do something, and observe neural activity in the brain when he or she does this, but that alone does not tell you what kind of code should be, written for the brain or for the robot equivalent. This is to assume we need to write any software at all. The software in the human brain, if there is any, seems to be already, hardwired into its structure. Therefore, if we somehow create a robot equivalent, there might not be any additional software needed. But on one of Moravec's scenarios for mind transfer, we have a super smart robot surgeon who instantly analyzes chunks of human brain, reverse engineers the code, and then writes the equivalent code for parts of the robot brain that are being created and installed as we go. Why do we need that robot? Apparently, it is too complex for a human to do. Where does that super smart robot surgeon come from? Or at least where did the knowledge come from of what code would allow a certain brain part to enable a certain human skill, and what equivalent code should be written for the robot, so that we can give such knowledge to the robot surgeon? If we know so much about the code, how did we arrive at that knowledge? On the other hand, if we know so little about the human brain that the super smart robot has to reverse engineer the code and then write the robot brain equivalent, then how did we create that super smart robot in the first place? Here, it seems to me, there may be the implicit assumption that the first method can come to the rescue--we already created that robot through the gradual process that overcame all the classic AI problems! At least we have to be such good coders by solving those classic AI problems and learning to build super smart robots gradually.

However, this seems to be going around almost in a circle. When we press for details about how we are going to do the first method, we get no reply. Though the authors do not explicitly say this, what we need is the help of the super smart robot that we will be creating, or at least the super smart robot created by the second method. When we ask about how we are going to do the second method, it is, assumed that we will be able to do it because we will have learned crucial details by already having done the first method. We may be exaggerating a little here to make this seem a circle, but not much. There do seem to be these two methods under discussion at the same time. Of course, some of this is to be expected, because we are really trying to blend the views of three different sets of authors, and they do not all say the same thing. However, no one really explains how each method is going to be successful without the results of the other.

To return to our questions, what would it show were a computer to pass the Turing test (in the original form or our more difficult amended version)? This is our second question. Passing the Turing test is not supposed to be a necessary condition of having or displaying intelligence (perhaps some intelligent things could not pass it), but it is supposed to be at least a sufficient condition of something. We will make three points about this. The Turing test is not a good test for our purposes because, first, strictly speaking, passing it really cannot be a sufficient condition of any kind of intelligence. Secondly, even if it were a good indicator of intelligence, it would be so only in a certain sense of “intelligence,” and third, if it really is to

replace the question of whether a computer can think we want it to resolve many questions we have about whether computers can think, and it does not do that.

One reason that passing the Turing test or any similar test cannot be a sufficient condition of intelligence is that it is logically possible to pass it by sheer chance. This is a trivial point but I have not seen it mentioned. A computer could through sheer chance, produce just the right responses during its trials with any number of guessers, say 100, that it could pass any test like this that you could throw at it. It is logically and physically possible that such a lucky computer could pass such a test for days on end through chance, but clearly, it would not be intelligent. However, this does not really show that the test is seriously flawed. In any kind of test we take in school it is logically possible for students to score a hundred percent through sheer guessing, even for tests that require a written response (guess the letters for the perfect answer). However, no one considers such tests flawed because of this reason. This way of passing a test is so highly improbable that no one worries about it, of course. However, it is not impossible, so passing the test does not absolutely rule out lack of intelligence.

Even if the above remote possibility does not preclude the Turing test from being a good test, there are other reasons to think passing it is not sufficient to indicate intelligence. One such reason is that a Block machine could pass the test and not be intelligent. A Block machine contains a great number of possible questions and plausible responses to those questions. When the Block machine is, presented with a question, it searches through its database and finds the question, and then randomly chooses one of the possible responses. When the guesser replies to that response, the Block machines goes through the same process to find a possible response, etc. Since the Block machine contains all possible questions (say, those up to 100 words in length), and numerous responses to these, the machine could fool the guesser indefinitely. The Block machine could pass the Turing test through this sort of brute force method, but since it clearly is not intelligent, so the objection goes, the Turing test is flawed. This may seems to be the same objection that I made above, but it is not, since a Block machine is not a random guesser.

Actually, though, what is controversial is whether a Block machine could pass the test, due to physical and technological limitations. Robinson claims that given the number of responses, the Block Machine would have to contain, and how fast it would have to be, operated to locate a plausible response in a very short interval of time, the machine could never really pass the test! If a fast enough machine could be, built, it would pass the test, but since such a machine is technically infeasible and cannot be, built, we do not have to worry about an unintelligent Block machine passing the test. Therefore, the Turing test could still be a good test for intelligence. (Note that the Turing test is supposed to be a test or criterion of intelligence, not a definition of intelligence.) Robinson's argument might be correct for current technology. One might try to argue that future technology might enable such a machine to be, built, but Robinson thinks not (Robinson, 1992) [37].

While Robinson claims the Block machine to be impossible, Copeland is not so sure that this means the Block machine example is irrelevant. Even if it were not technically possible to build such a test-passing Block machine, the fact that it seems theoretically possible for a Block machine to pass the Turing test would show that the test is not quite right. I tend to agree and think this is another reason to think the Turing Test is not a good test. Copeland calls the Block machine possibility the “black box objection.” What matters for intelligence is not only that the Turing test is passed, but how it is passed, and the “how” should become part of the test. The original version of the test considers conversational output as the only

criterion. Copeland thinks this output criterion needs to be, supplemented by a design criterion, which come in two versions. One version is that the program should do the things it does in a way broadly similar to the way those things are done by the human brain. In other words, there should be, a high-level equivalence between the ways the computer does things versus the ways the human brain does them. The equivalence does not have to go all the way down. At this high-level, for instance, one could say that a modern computer and an old valve machine are executing the same program (Copeland, 1993, pp. 50-51) [38].

This requirement for the computer to be of anthropocentric design is the strong version of the design criterion. Copeland defends it because, first, the Turing test is not supposed to be a litmus test for thought, specifying only a sufficient condition, not a necessary one. If chimpanzees and some computers fail, this does not show they cannot think. Second, Turing's test is anthropocentric anyway in that it specifies the test for thought in terms of producing outward behavior indistinguishable from that of a human, so the strong version of the design criterion is in keeping with this spirit (Copeland, 1993, pp. 50-51) [38].

We are not sure what to make of this strong design criterion. Whether or not we make the Turing test this strong depends on how strict we want to be about the term “thought” and our test for it. We run the risk of excluding some intelligent beings who do not think as humans do. Should we allow the term “thought” to apply in such a case? This question will reappear in a different form, when we consider the mind-body problem. When we wonder whether we wish to attribute mental states to a robot on the basis of their output being similar to that of a human (functionalism) or allow that they don't have particular mental states unless the robot brain states are the same as human brain states (type-identity materialism). The general view of our authors seems to be that the robot brain will run in a roughly similar fashion to the human brain (parallelism), so perhaps it will pass this strong version of the test. It depends on how far down the similarity has to hold. We have seen above that though our authors have not been explicit about the degree of parallel processing, what they have in mind may or may not hold too far down, depending on the scenario. In addition, their obsession with speed may be an indication that they do not necessarily envision robot and human brain similarities holding very far down at all.

The weak version of Copeland's design criterion is that the program or computer must be of modular design. This means that it must in principle be capable of being “zipped whole” and piggy backed onto other programs, for example, those running the sensory systems of a robot, with the new whole forming a functioning program of greater complexity. This is weaker than the insistence that the program operate analogous to a human brain, but it is strong enough to guard against passing the Turing test through trickery along the lines of a Block machine (Copeland, 1993, p. 51) [38]. The robots envisioned by our authors in the extraordinary future probably would pass this requirement.

Our second point above is that even passing the Turing test would indicate only certain kinds of intelligence. Let us suppose that a computer passes the test, and passes it by operating a certain way analogous to the way humans operate (and not like a Block machine). This may indicate only that the machine has a particular kind of intelligence, or intelligence in a certain domain. The Turing test is all about conversational ability. If intelligence is as broad as Gardner thinks it is, and then some kinds of intelligence may not be, tested for adequately using the Turing test. As we already questioned, how would this test determine the extent to which one was a dancer, or had any bodily-kinesthetic intelligence at all, for instance? It seems a disembodied computer might be able to converse about dancing, but that does not

seem to be what Gardner means by bodily-kinesthetic intelligence. In this case passing the test would not be sufficient to indicate the requisite possession of the relevant type of intelligence. The ability to talk about dance is not the same as the ability to dance, so instead of trying to make the Turing test be all things for all purposes, perhaps one should consider another type of test. I guess one cannot fault Turing for failing to consider this--the kind of intelligence he had in mind was associated with thinking and the use of language. (In fact, he never even claimed the Turing test was a test of intelligence, but rather a substitute for discussions of whether the computer could think.) However, Gardner thinks all humans possess all kinds of intelligence to some degree, so what we want is a test such that passing it is sufficient to indicate the possession of all types of intelligence, and the Turing test does not do that. I am not arguing here that Gardner is correct, but we do not want to rule out his position from the start, which we would do if we used an intelligence test that could not test for all the types of intelligence he mentions. Maybe we need to supplement the Turing test with another test.

Our third point about the Turing test is that it is not clear that it is a good test for our purposes because if it really is to replace the question of whether a computer can think we want it to resolve many questions we have about whether computers can think, and it does not do that. For instance, even if a computer passes the Turing test in the right way, it may be that we still will consider open such questions as whether it is consciously thinking, or whether it "really" has original intentionality. To consider the objection we need to more fully discuss some distinctions related to computers and thinking.

As humans, we commonly take ourselves to engage in the activity called "thinking," which might be taken to include matters such as believing something, desiring something, and so forth. We take ourselves to engage in conscious thought, and many people believe we also engage in unconscious thought. Turing apparently did not believe it was fruitful to continue to discuss the issue of whether a computer could think. His suggestion that we should consider this question to be resolved affirmatively in the case of a computer that passed the Turing test might indicate that one should take "thinking" to be the carrying out of a certain function on the part of the thinker. On the other hand, one might take, "thinking" to either or as well refer to the certain inner states of conscious awareness or the like. We will discuss this a little more in the next chapter of this thesis.

However, one thing I want to address here is what it might mean to say that whether a computer thinks is an issue that calls for a decision rather than a discovery. It is not clear to me what this means because it could mean any of several things.

For example, Copeland argues that the question of whether a computer can think can remain open even after all the relevant facts are, known because it is an instance of an application of a concept to a relatively new sort of case. This sort of case was not, envisaged when the concept was framed. Our concept of thinking was, formed in the context of application to natural living organisms rather than artifacts. What we need to do is decide whether the notion of a computer thinking best fits the purposes for which the concept of thinking is employed (Copeland, 1993, pp. 52-54) [38].

What Copeland seems to mean is that "thinking" is linguistically indeterminate (for want of a better phrase) when applied to computers. However, there seems to be other kinds of indeterminacy that might be, meant when it is said that a decision rather than a discovery is needed. To say that a decision is called for rather than a discovery suggests either that (a) the issue is metaphysically indeterminate, (b) the issue is metaphysically determinate but

epistemologically indeterminate, or (c) the issue is metaphysically and epistemologically determinate but linguistically indeterminate. To illustrate my use of these phrases, consider the question of whether “Katie is imagining a pink elephant” at a particular time (assume we agree about which person named “Katie” we are talking about). Suppose we say the issue of whether Katie is imagining a pink elephant calls for a decision rather than a discovery. What could I possibly mean? (a) If what I have in mind is that the issue is metaphysically indeterminate, I mean that though we all agree on the meanings of the terms “pink,” “elephant,” “imagining,” etc., and we can ask Katie whatever we want, with nothing lacking in the way of her answering our questions, there is no fact of the matter about the issue. Therefore, it could be, argued, this situation would call for a decision rather than a discovery. Nevertheless, this is not how we usually think about things in the world. In everyday life, we would claim that the question about Katie is metaphysically determinate; it is a fact that she either is or is not imagining such a thing. (b) If I say that the issue is metaphysically determinate but epistemologically indeterminate, then I mean that though we all agree on the meanings of the terms involved, and though there is a fact of the matter about whether Katie is thinking of such a thing, we cannot find it out. Perhaps we are not sure whether to believe her answers. In such a situation, it might be, said that since a discovery of the truth is not possible, a decision is called for. (c) If I say that the issue is metaphysically and epistemologically determinate but linguistically indeterminate, then I mean that though the physical facts about Katie are all fully determinate, and we can know what these are, we are just unsure about whether the term “pink” or perhaps “elephant” applies in this case. Here too it might be, said that a decision, rather than a discovery, is called for.

So in what sense does Copeland think that the question whether a robot thinks calls for a decision rather than a discovery? His comments above seem to indicate that he thinks the issue is linguistically indeterminate, at least until we decide that we can apply the phrase to computers. To see this, note some further comments he makes about the concept of thinking. Copeland considers three possible uses for the concept of thinking. The first is to pick out entities having an “inner awareness.” Involved here is a phenomenological distinction. However, Copeland thinks that this is not a good definition of thought because of the coherence of the notion of unconscious thought, which is, thought without that inner awareness. (We are not objecting here to Copeland's view that humans do engage in unconscious thought.) The second possibility is for “thought” to refer to a biological distinction, with it picking out those organisms with higher brain processes. However, this will not work because there could be extraterrestrial thinkers having no such brain processes. (More is involved here than Copeland lets on, such as whether one's position on the relation of thought to brain processes is that of a functionalist or a type-identity materialist, but this discussion will have to wait until a later chapter.) The third possibility, which Copeland likes, is that thought distinguishes organisms with flexible inner processes allowing plasticity in response to the environment from those whose behavior is instead rigid. The former have processes that are “massively adaptable,” and they can “form plans, analyze situations, deliberate, reason, exploit analogies, revise beliefs in the light of experience, weigh up conflicting interests, formulate hypotheses and match them against evidence, make reasonable decisions on the basis of imperfect information, and so forth” (Copeland, 1993, pp. 55-56) [38]. Now in line with his earlier rejection of the first option, presumably these inner processes need not be conscious. This understanding of the term “thought” seems to characterize it partly as the possessing of a kind of function or capacity, though not entirely

solely in these terms, since Copeland does mention that it involves having inner processes that provide for such functionality.

Copeland thinks that probably the most important role for the concept of thinking is in the explanation and prediction of behavior. Someone doing something is explained on the basis of that person having thought such and such rather than in terms of electrical or biochemical activity in the brain (though if type-identity materialism is correct, then such thoughts just are brain processes). Such explanations are intentional explanations. If AI research were to produce massively adaptable programs in robots that roughly matched those of humans, we could apply intentional explanation to them, and such robots could be, said to think. “It is clear, then, that the purposes for which we employ the concept 'thinking' would be best served by a decision amongst the linguistic community to count these robots as thinking things,” literally, and not just metaphorically (Copeland, 1993, pp. 56-57) [38].

We agree that if thinking in terms of having states of phenomenal consciousness is, left out of the picture, we could say that such robots (if and, when they are built) think, in his sense of thinking. (We do not know that they do have phenomenal consciousness, and we do not know that they don't, as I will argue in a few chapters.) However, it might not be entirely clear why a decision rather than a discovery is called for. It would seem that whether such robots think would be a (metaphysically) determinate matter, given his definition; they either have flexible inner processes allowing plasticity or not. If thinking does not have to involve conscious awareness, then these robots could be unconscious or nonconscious zombies and still be thinking. And if it is a metaphysically determinate matter, then we should seek to discover whether they do such thinking. The reason he holds that a decision is called for cannot be that we will never agree on what the terms mean (linguistic indeterminacy), because he just defined them. What Copeland seems to mean is that the term “thinking” is currently linguistically indeterminate, but it will cease to be so after he defines it in the context of robots? Then would not the issue, once defined, call for a discovery rather than a decision? So both decision and discovery might be involved after all.

Another matter Copeland is not clear on is whether he thinks intentionality involves conscious awareness--it looks like he holds it does not. If it does not, then perhaps we could make sense of the concept of robot intentionality, in the relevant sense of “original” intentionality. The claim is sometimes, made that robots could have only derivative but not original intentionality. Intentionality characterizes human mental states and refers to the fact that they are about something. For example, my belief is about a book, a person, an airplane, etc. My belief is not about my own mental state but about something in the world. This is original intentionality. However, consider the words and sentences in a book. Are they about anything? They are only if we make them be about something. In other words, the sentences in the book are not about anything from the perspective of the book--the sentences are not the book's beliefs; the book has no beliefs to be intentional. Any intentionality of the sentences in the book must be derivative on some agent who has original intentionality.

Well, the question arises as to whether the symbols of a computer have any original intentionality, which is the question of whether from the perspective of the computer they are about anything. It seems that to have original intentionality, the computer would have to be a subject. Then its symbols could represent external matters to the computer. One might also try to argue that to have original intentionality the computer would have to be phenomenally conscious, though this is more controversial. If we can make sense of the notion of

unconscious thought, then we might be able to make sense of the notion of something being a subject and engaging in intentional behavior without it being phenomenally conscious.

One view is that in an important sense the computer really is just like the book: the computer's symbols would be about something only if we interpreted them as about something. That is, any intentionality would be derivative on our original intentionality. The computer has no ability to give its symbols any representational meaning. (As we shall see, this is Searle's view, which may or may not depend on a view that computers are not conscious.) A different view is that the computer, if it attains subjectivity (whether or not conscious), can give its symbols intentionality because they can represent things in the world to the computer. They can have meaning, even if the computer is not conscious. Not all meaning has to be conscious meaning. Copeland seems to take this latter view.

We think that even if a computer passed the Turing test, even in Copeland's amended version that takes into account the way it is, passed, we might still have questions about thinking that could be said to call for discovery rather than mere decision. We might allow that the computer could think in Copeland's sense, and yet still wonder whether its thinking is conscious and whether its thinking involved original intentionality. I see no reason to believe "a priori" why any type of indeterminacy should be, assumed to hold in the case of computers any more than it does in the case of humans. So one might plausibly claim that there would still be important matters, possibly metaphysically and linguistically determinate but epistemologically indeterminate, that called for discovery. It seems the only reason he could have for saying a decision is called for would be if we thought we just would not be able to discover the truth and for practical purposes had to decide. Nevertheless, depending on our view of the relation of the brain and mind, maybe we could decide. If thoughts are just brain processes, for example, (a "type identity" version of materialism discussed later) then could not we discover whether, or not the robot brains in question had the relevant processes?

Turing considers his Turing test a substitute for the issue of whether a computer can think. With respect to Copeland's third sense of "thinking" above (massively adaptable and possibly unconscious processes of reasoning, etc.), it may appear to be a good test. However, even here, we seem to have to modify it to incorporate the matter of how the computer passes the test and it does not resolve questions about in what sense computers are intentional or conscious. In the sense in which "thought" can mean "conscious thought," passing the test may have no bearing on whether a computer thinks. It may be possible for something to pass the test and yet not be conscious at all in the sense of phenomenal consciousness. "Phenomenal consciousness" will be, discussed more fully in a later chapter, but, basically, it is the phenomenal awareness of things like pain that you have but an unfeeling zombie would not. Copeland above allows that thought could be unconscious, which may be true, but his argument is irrelevant to the question of whether the test is a good test for thought if what Turing meant was "conscious thought." However, even if that is not what Turing meant, we would still want to know that a robot is conscious before transferring our mind into it. Therefore, if the Turing test is, trotted out as an appropriate all-purpose test to give to a robot to test for what you want the robot to have in the way of a mental life, then we do want it to test for consciousness. A robot who thinks unconsciously could still pass the Turing test and not be conscious, and this makes the test not what we want.

We argued above that it is not clear when, if ever, a computer will be able to pass an appropriate version of the Turing test, and that even were this to happen, we still might not be satisfied that we have no need to further discuss the issue of whether it can think. John Searle

also claims that the Turing test leaves something to be, desired as a test of machine intelligence and thought.

10.11. COMPUTERS AND UNDERSTANDING

John Searle claims that no matter how intelligent computers appear to be, they are not truly smart in any way analogous to humans. In a famous line of argument involving a Chinese translation room, John Searle objects to the claim that computers will ever be able to understand anything. Even if a computer were to pass the Turing test, it would not understand anything it does. The implication is that a machine that can understand nothing is not intelligent in our sense of the term.

Searle attributes to strong AI the claim that an appropriately programmed computer really is a mind; the right computer with the right program can understand and have other cognitive states (Searle, 1980, p. 353) [39]. He examines the work of Roger Schank, whose computers are, provided with “scripts” that fill in background, common sense information about various situations and scenarios. We have already seen the need of robots for such background knowledge. This information enables the computer, when presented with a story involving such a situation or scenario, to “infer” things it has not been explicitly told or that strictly speaking cannot be deduced from just what it has been told. Strong AI proponents, according to Searle, hold that the computer is not merely simulating human cognitive abilities. They would claim the computer literally understands the story and that the way the machine understands here explains how it is that we understand such a story. Searle thinks both claims are false (Searle, 1980, p.354) [39].

Strong AI holds that the human mind works on principles that are operative in such computer “understanding,” so Searle invokes a thought-experiment to show what it would be like if our minds really did work like strong AI claims they do. Searle's thought experiment involves the Chinese language, about which he knows nothing. Searle is locked in a room and supplied with Chinese writings (the script) to translate, another batch of Chinese writings (the story), and rules (in English) for formally relating (by the shapes of the letters) the second batch to the original Chinese writings. He is then given a third batch of Chinese symbols (the questions) and English rules that tell him how to correlate the third batch with the first two and give back Chinese characters in response to the third batch. The Chinese characters he gives back are, considered the “answers,” and the English rules are the “program.” He gets so good at doing this “translation” that his answers are like those of a native Chinese speaker. He is also, given similar writings and questions in English, and since he understands English, he has no trouble answering these questions (Searle, 1980, p. 355) [39].

Searle claims that it is obvious he does not understand the Chinese stories, even though his answers are on par with his answers to the English scenario questions. The English he understands, but in the case of the Chinese, he merely manipulates uninterrupted formal symbols. In this regard, performing computational operations on formally specified elements, he behaves like a computer and is the “instantiation” of a computer. So likewise Schank's computer does not understand, its stories, since the computer has nothing more than Searle does when he understands nothing. Furthermore, there is no reason to think this type of symbol manipulation serves to shed any light on how humans do understand. What is going

on in the computer, as in the case of the Chinese “translation,” is not sufficient for understanding, and it has not been, shown to be a necessary or even a significant contribution to understanding either. If he were to do a similar procedure with English characters, he would understand what they meant, but this is not the case with the Chinese. The computer can process characters syntactically, but it does not know their semantic meaning, and so it has no understanding, much in the same way that he would have no understanding of the Chinese characters no matter how good he became at this “translating.” So when he really does understand the English story, the claim of strong AI that he is really just doing more of the kind of thing the computer does and what he himself does in the Chinese case is “incredible.” He admits, though, that he has not shown this claim to be false (Searle, 1980, pp. 355-356) [39].

In the Chinese case he has everything a computer does, and yet understands nothing, so there is no reason to think that his understanding in the English case has anything to do with computer programs (computational operations on purely formally specified elements). The example, he thinks, suggests that such operations have “no interesting connection with understanding.” No reason has been, given to show that in understanding English he is operating with any formal program (Searle, 1980, pp. 356-357) [39].

Searle points out that the issue here has nothing to do with any vagueness in the concept of “understanding.” Whatever confusions there may be about borderline cases, there are two absolutely, clear cases here of understanding (the English translation) and not understanding (the Chinese translation) (Searle, 1980, pp. 357-358) [39].

Searle already tries to fend off certain types of criticism in the original article by considering several possible replies to the argument. The systems reply claims that though Searle does not understand, he is only part of the system, and the system understands. Searle replies that even if he were to internalize the elements of the system by memorization he still would not understand. “If he doesn't understand, then there is no way the system could understand because the system is just part of him.” However, he thinks, the systems reply is absurd anyway, for how is it that Searle alone would not understand but the conjunction of Searle and bits of paper would understand? It could be claimed that the man is really two formal symbol manipulation (sub) systems, one understanding English, and the other understanding Chinese. Nevertheless, in the case of subsystems, Searle claims the one manipulating Chinese is no better off than was the man himself. One can even imagine the Chinese subsystem passing the Turing test, which shows that passing the test is not sufficient for having understanding (Searle, 1980, pp. 358-360) [39].

The robot reply holds that what is needed is to put a computer in some kind of robot body so that it has “perceptual” input from the world and “acts” in the world (behavioral output), rather than just having the input and output of formal symbols. Obviously, this is what the extraordinary future is supposed to bring. Searle replies on several levels. First, as pointed out by Fodor, this would tacitly concede that cognition involves causal relations with the outside world rather than being just formal symbol manipulation. Even so, putting a computer in a robot body adds nothing significant in the way of understanding or intentionality. If Searle and his room were in the robot, so that some of the messages from outside came via the television camera and some of the symbols he gave out moved robot arms and legs, there would have been nothing added to create understanding where there was none before. “The robot has no intentional states at all,” and by instantiating the program Searle would not have any intentional states of the relevant type. He is still merely manipulating symbols, and it is

still syntax without semantics (Searle, 1980, pp. 362-363) [39]. Note that Searle here links intentional states to understanding.

The brain simulator reply invokes the example of building something to simulate the actual sequence of neuron firings at the synapses of the brain of a native Chinese speaker when he understands and responds to the story in Chinese. This system even operates by parallel processing. Searle notes this to be an odd reply for AI. Strong AI takes the mind's relation to the brain to be as software is to hardware, with the essence of the mental to be computational processes over formal elements. The whole idea is that we do not really have to understand how the hardware works if we know the software, but here in this reply we have to recreate in detail the low level workings of the brain. Even so, to Searle the reply fails. Add to his Chinese translation story a series of valve water pipes and connections analogous to neurons and synapses, with the man opening and closing valves in response to instructions. The output pops out as water from the pipes if all the right faucets are, turned on. This simulates the formal structure of a Chinese speaker's brain, but the man does not understand, the water pipes do not, and neither does the conjunction of the man and the water pipes even if the man were to do all the firings “internally” by imagining the water pipe operations. The simulation is of the wrong brain properties. What we need is not the formal structure of the sequence of neuron firings but the causal properties that produce intentional states. The example shows that the formal properties are not sufficient for the causal properties (Searle, 1980, pp. 363-364) [39].

The combination reply claims that understanding would come from putting all of the previous replies' aspects together. A brain-shaped computer is in a robot body, and it is, programmed with all the synapses of the human brain. Its behavior is indistinguishable from that of a human. Searle thinks we would be tempted, and find it irresistible; to attribute intentionality to it, but this would be for reasons irrelevant to strong AI. If we knew nothing more about the robot, we would assume it had intentionality because of its looks and behavior. However, this would not show that instantiating a formal program was constitutive of intentionality. Once we found out how it worked, the game would be over, and we would see it as analogous to an ingenious dummy (Searle, 1980, pp. 364-365) [39]. Contrast this view with that of Copeland above, who was willing to grant intentionality based on external behavior of sufficient complexity.

Searle considers other replies that we are not going to recount in detail. The other minds reply claims that we should allow the same claims about the computer as we do to other humans. When we attribute minds to them, since we have similar evidence from their behavior, but Searle thinks this confuses the epistemological issue of how we know about other people with the metaphysical issue of what cognitive states really amount to. The many-mansions reply is that in the future we might be able to create different computers that will understand (though we don't know how these might work), but to Searle this is just to redefine AI in an ad hoc way as whatever in the end will produce real intentionality. This shows nothing about the current claim that is under consideration (Searle, 1980, p. 366) [39].

At the end of his famous article Searle tries to state his position more explicitly with respect to a number of issues important to us. First, Searle is not arguing against materialism or the view that machines can think. For one, he thinks we are machines that can think. It is an empirical question whether there could be an artifact, a man made machine, that could think. There is something about Searle or any human that the computers in question lack that allows him to understand English--why not give this to the computers? If we were able to

build a machine out of the same material of which we are made (neurons, etc.), then we would have duplicated that in us which has the requisite causal powers and so it would obviously think. (Note here that Searle has shifted talk from “understanding” to “thinking” and seems to mean conscious thinking). However, if the question is whether something could think or understand solely by virtue of being the right sort of program, the answer is no. This would not be likely, where the operation of the machine is, defined solely in terms of computational processes over formally defined elements (the instantiation of a computer program). It is not because he is an instantiation of a computer program that Searle can understand English and have other intentional states. It is because of his biological structure, and it is this that would have to be given to the computer. Only something that has the causal powers of this structure could have intentionality. It is an empirical question whether other things like Martians could be made of the kind of stuff that has such a structure. What matters about the brain is not the formal sequence of synapses but the actual properties of the sequences. The formal symbol manipulations of computers are not even real symbol manipulations, since they do not symbolize anything. Any intentionality involved is in the minds of the programmers and interpreters (Searle, 1980, pp. 367-369).

Searle thinks there are severe problems with the assumption that the mind is to the brain as a computer program is to the computer hardware. The same program could have many different realizations in different entities and systems that have no intentionality. He mentions Weizenbaum's example of how to construct a computer out of toilet paper and stones, which to Searle clearly would lack intentionality even though it could instantiate a formal program. Likewise, for his Chinese, translation system using water pipes. In addition, intentional states are not just purely formal in the way a program is. Intentional states have content--the belief that it is raining is not, defined as a particular formal shape but as mental content with conditions of satisfaction. Also, mental states and events are a product of the operation of the brain in a way that a program is not a product of the operation of the computer (Searle, 1980, p. 369) [39].

People have been, fooled by the notion of simulation into thinking a computer simulation of an event or process is the real thing, but computer simulations are not, confined to simulations of thinking. “No one supposes that computer simulations of a five-alarm fire will burn the neighborhood down or that a computer simulation of a rainstorm will leave us drenched. Why on earth would anyone suppose that a computer simulation of understanding actually understood anything?” Searle notes that it is sometimes, bemoaned that it will be difficult to get a computer to feel pain, or fall in love, but these would be no harder than cognition or anything else. Simulation just involves the transformation of some input to output by a program, but this is all the computer does with anything, even a simulation of thinking. The assumption is, based on a confusion of simulation with duplication (Searle, 1980, pp. 369-370) [39].

Searle accuses strong AI of assuming a version of dualism. This is not a substance dualism, but rather the view that the mind is really, independent of the brain in the sense that it could be, realized as a system of computational symbol manipulation operations in any number of media, just as programs are independent of their realization in machines. Strong AI workers often rail against dualism without realizing that they have their own form in holding that what is specifically mental about the mind has no intrinsic connection with the actual properties of the brain (Searle, 1980, pp. 371-372) [39].

Searle's original paper generated a storm of controversy. Some readers agreed with Searle and considered his argument decisive--digital computers made of silicon will never be able to think, at least not in virtue of their formal symbol manipulation. Other readers, including many in AI, thought Searle guilty of some sort of illusion. The claim was that his thought-experiments were completely unrealistic and therefore led his readers to gloss over important leaps in the line of reasoning--and this sleight of hand vitiated the force of his argument. Furthermore, there seems to be controversy over who has the burden of proof. Does Searle succeed if he shows that we cannot assume that machines (in the sense of modern digital computers) can think? On the other hand, does he actually have to prove they cannot? Do Searle's critics succeed if they show that Searle has not proved that machines cannot think? Alternatively, do they have to prove that machines can?

As a way of appraising Searle's views, we want to distinguish between Searle's arguments and Searle's position. Of course, Searle's arguments are, intended to convince you of the correctness of his position, but they are distinct. As I will discuss below, his arguments have been severely criticized, perhaps effectively, but even if his arguments are, undermined we will still want to consider whether his position is correct. Part of what we are saying here is just the well, known point that the invalidity of an argument does not prove the conclusion false. We may put forth the most comically fallacious argument to prove to you that George Washington was President, but the fact that the argument is invalid does not prove that Washington was not President. So in the comments that follow, we will discuss Searle's arguments, but in the end we are more interested in whether his position is correct than whether his arguments work. In addition, we are not really, interested in reconstructing the historical Searle either. If his comments lead one to a particular interpretation of his argument such that the argument is invalid, I would be interested in finding out whether his claims and position could be, reinterpreted as a different argument that might be insightful in its own right.

Much ado is made about how what Searle describes--the massive project of translation by a single individual--would be impossible to pull off. It would take days, weeks, months, years, etc. of effort. Similarly, the variation on the story that involves the individual memorizing of millions or billions of rules describes a task that could never be, done (for example, see Hofstadter, 1981, pp. 373-375) [40]. We do not see that the fact that the scenarios in the thought experiments describe unrealistic situations automatically entails that Searle's reliance on them is faulty. The extremely common philosophical use of thought-experiments usually assumes only that the depicted scenarios be logically possible (not violating the laws of logic and thus inherently self-contradictory, for example), not that they be physically possible (obeying all the laws of physics) or technologically possible (possible given current technology). So it is not obvious to me why Searle's thought-experiments have to obey restrictions not followed by countless bizarre body, brain, and mind-switching experiments and instances of teletransportation and brain fission depicted in the literature on personal identity, for example.

Let us look at some specific criticisms of the Searle's argument, particularly first with respect to the systems reply type of objection and then later with respect to the robot reply type of objection. Copeland thinks Searle's argument that the whole system does not understand is fallacious. (Copeland discusses the version involving Sam the story understanding program, but his remarks would apply to Searle's earlier versions as well, since Searle has never changed the essentials of this critical line of reasoning). However, Copeland

thinks Searle is focusing on the wrong participant in his Chinese translation scenarios. Sure, the person doing the translation does not understand Chinese, because he is merely a laboring cog in the whole process. However, if you could ask the voice of the whole system or program (or process of translation), this voice would reply that she understands what the words and sentences mean (whether or not she does, she will say that she does). Searle moves fallaciously from the assumption that no amount of symbol manipulation by a part of the system will enable that part to understand the input to the conclusion that no amount of symbol manipulation by a part of the system will enable. The wider system of which the part is a component allows us to understand the input (Copeland, 1993, p. 125) [38]. Moreover, of course strictly speaking such a conclusion does not follow from this premise. Copeland is correct that if this is the whole of the argument it is deductively invalid. You cannot legitimately infer from the fact that part of something has a particular property that the whole has that property. This is an old logical fallacy known as the "part-whole" fallacy. Copeland assumes Searle has the burden of proof and his opponents win if they show his arguments are faulty. Moreover, if his argument is intended as a deductive argument, and it is invalid, then it is faulty.

As Copeland points out, Searle explicitly considers this part-whole distinction under the rubric of the systems reply and thinks it silly to claim that the part alone doesn't understand but the part conjoined with slips of paper do understand. (Here again we have the burden of proof issue. Is Searle's opponent claiming only that the whole system may understand (Searle not having shown that it cannot) or that the system does understand (and then where is the argument for this?)?) The critic claims the silliness is not from the notion of the whole system understanding but because Searle's scenario is so ridiculous--a man trying to translate by manipulating untold numbers of rules and slips of paper (Copeland, 1993, p. 126) [38].

However, let us take the comments and reinterpret the debate. If the strong AI proponent is merely asserting that the system understands, then this does seem to be question begging, as Searle argues and Copeland agrees (Copeland, 1993, p. 127). But let's be more charitable than that to strong AI and take it that the strong AI proponent is claiming that we can assert that computers can or will think by using an analogy with humans. Just as we attribute thought to other humans based on their having features and behavior that are relevantly similar to us (though we cannot "be them" to really prove to ourselves that they think). So likewise, we can take it that computers think if we see such computers exhibiting relevantly similar behavior (like passing the Turing test, or in robot bodies conversing with us). This seems to be an argument from analogy on the part of the proponent of strong AI, and though it is not strictly speaking deductively valid, it might be, accepted as a decent inductive argument (if the analogy really holds). Copeland seems to agree that strong AI might be using this strategy: "In my view it is as obvious as anything can be in philosophy that if we are ever confronted with a robot like Turbo Sam, we ought to say it thinks," and, "there can be no point in refusing to say he understands the language he speaks. For in my imaginary scenario Turbo Sam is every bit as skilful as a human in using words to ease and ornament his path through the world" (Copeland, 1993, p. 132) [38]. In addition, I do think that strong AI seeks to build an analogy between humans and computers--proponents think that the way a computer works sheds light on the way the human mind works.

Searle can be, interpreted as calling into question such an analogy. He is pointing out that the analogy holds among humans, since humans are made of the same kind of stuff. However, computers, built out of different stuff, are not similar in this respect, and thus there is a

disanalogy between humans and computers. Therefore, the strong AI attempt to show computers can or will understand by using an argument from analogy does not succeed.

Why should we think the kind of stuff involved is relevant and breaks the analogy? Because we can conceive of a situation (the Chinese room), in which symbol manipulation would not seem to produce understanding purely by virtue of the symbol manipulation. This seems to me to be the real merit of the Chinese translation examples--to try to break the analogy between humans and computers. Searle thinks that the Chinese translation scenario describes a situation that is relevantly similar to what goes on in computers and shows that if humans worked like computers, they would not understand. If humans operated like computers, and “understood” purely based on symbol manipulation (the thesis of strong AI), they would not really understand. Nevertheless, we do understand, so our understanding cannot come from doing what computers do and computers are not enough like us to be analogous. Of course, Searle appeals to our intuitions at this point. The translator does not understand and neither does the whole that includes the translator and the slips of paper. I must admit that Searle's scenario strikes me as powerful and effective, since I have trouble believing the Chinese room system understands--while I can more easily wonder whether a future supercomputer understands, my intuition is that the system he describes in the Chinese room experiment does not understand. In addition, does anybody not trying just to win the argument seriously think the system he describes would understand, though the translator would not? Very few people would go as far as to really, think the Chinese room system understands. Churchland does not, and he has been one of Searle's major opponents over the years. Even Copeland, who thinks Searle's argument is wrong, admits that this type of system (in the form of the Sam program) would not understand (Copeland, 1993, p. 128) [38]. So why is Searle wrong to appeal to this intuition? If he thinks he has a good argument that proves that the system would not understand because the part does not, he is wrong. Nevertheless, if what he is doing is showing that the relevant analogy with a computer is a Chinese room and not a human, and appealing to our intuition that the Chinese room would not understand, we do not see that his position is all that faulty. If we agree, then Searle has broken the analogy between computers and humans, and the Chinese translation scenario does seem relevantly analogous to computer processing. Those who would argue otherwise would claim that the Chinese translation situation is not analogous to computer processing. Just because, computers would process faster? Because the individual involved would become exhausted? Searle thinks that these issues are irrelevant, and I personally fail to see that such things show the situation is not relevantly analogous to computer processing. Speed and resource exhaustion seem irrelevant to symbol processing per se. Strong AI holds that understanding is a function of computation, not computational speed per se or whether parts of the computer become “exhausted” during the processing. To recap, the similarities on which strong AI bases the analogy between humans and computers are first, functional output in response to input, and second, information processing by symbol manipulation. Since in the Chinese translation room scenario both of these occur but without the presence of understanding, and since in humans there clearly is understanding, there seems to be a respect in which computers and humans are relevantly different. Any reliance on an argument from analogy on the part of strong AI is therefore, called into question.

But it does seem to come down to one's intuitions about the Chinese room scenario. Few strong AI proponents seem willing to allow that thermostats, or smoke detectors or even today's PCs literally understand, but somehow when the computers get more sophisticated

then they start understanding. (And to win the argument they will claim that, sure, the Chinese room system understands.) Searle's position is that nothing relevant would have changed between today's machines and those in the future, and so the disanalogy with humans and computers would still hold. However, if your intuitions are that today's PCs do understand, and you really think that the Chinese room scenario the man conjoined to the bits of paper would understand. Alternatively, if you think that processing speed really does make the relevant difference, then you will not buy the claim that the analogy between humans and computers has been broken and Searle will not succeed with you.

On the other hand, Searle's amazement that people mistake the simulation for the reality is a little hard to swallow. Searle claims that while a real fire is hot, no one takes a simulation of a fire to be hot, and so likewise no one should mistake a simulation of thinking for actual thinking. The reality and its simulation are distinct, but is it really, so surprising that one could think that a simulation of thinking is thinking when it (on the hypothesis of a functioning robot) results in the same type of output for input? If you feed real firewood, it produces heat, and if you fed a simulation of a fire real wood, and as output, you got actual heat, you might think that the simulation was as good as the reality and in fact was a fire. So when you talk to a robot and it responds in exactly the same way that a real person would respond to that input, then you might say that this simulation of thinking is as good as the thinking it simulates and might really be the same thing.

Searle has not retreated, despite much criticism (and some misunderstanding) from the AI community. In a recent book, he reiterates his earlier point (and of course claims he refuted strong AI with his original argument). He says, "If I do not understand Chinese solely on the basis of implementing a computer program for understanding Chinese, then neither does any other digital computers solely on that basis, because no digital computer has anything I do not have (Searle, 1997, p. 11)." The structure of the argument is, summarized as:

1. Programs are entirely syntactical.
2. Minds have semantics.
3. Syntax is not the same as, nor by itself sufficient for, semantics.

Therefore, programs are not minds. Q.E.D. (Searle, 1997, pp. 11-12) [41].

We are sympathetic to Searle's claim that it cannot be in virtue of the sheer fact of computation that we understand. It must be in virtue of something else. He thinks that this something else has to do with the causal powers of the brain. Presumably, if it is not in virtue of the sheer fact of computation in the case of the brain, then of course it must be something else, and if one wants to call that mysterious something, else the "causal powers" then I have no objection. However, Searle leaves the impression he thinks the causal powers have something to do with the fact that the brain is wetware rather than hardware. We certainly do not know what this something else is, or that it is some property of wetware per se. It might have to do with the organization, structure, or mode of computation of the brain rather than just the sheer fact of computation itself. Surely, there is the possibility that this could be, recreated in a computer. Searle recognizes that he has not shown that future robots will not understand, only that if they do it will not be in virtue of the sheer fact of computation over formal symbols that they do so.

We are also sympathetic to a more general point that we can learn from Searle's comments, namely that there is a lot going on in the human brain and mind and we should not

just assume that computation captures all of it, assuming it even captures any of it. We can do calculations, but this doesn't mean that a calculator captures what is going on in the brain or mind when we calculate, much less perceive and appreciate a sunset, love another human being, feel joy at the birth of a son or daughter, etc. To assume that all the many subjective experiences of humans are, captured in the notion of computation seems naive.

If my interpretation or reinterpretation is correct, then though Searle has not shown that the system does not understand, he has called into question the analogy on which strong AI builds its case that the system does understand. However, the above discussion focuses on the systems reply. What of the robot reply? Maybe the reason that the man, or the system, in the Chinese translation scenario fails to understand is that he (or the system), like isolated computers, has no way of connecting to the external world. This is the essence of the robot reply to Searle.

Recall that Searle's response to the robot reply is twofold. First, to claim that robot interaction in the form of input and output involving the external world is necessary for understanding is to concede that computation alone is insufficient. This seems correct, but it may be attacking a straw man. Strong AI proponents do claim that sophisticated computers will think and understand and do so in the way we do. However, not all would claim that anything that computes thinks--the amount and type of computation may be relevant, or other organizational factors from the brain. The sheer fact of computation is not enough. So many might already grant that in some sense, computation alone is not enough--it must be a certain kind of computation, etc. Someone like Schank, at least, does not think his early programs were thinkers, though he saw potential future versions of this type of program as thinking. But it might be argued by Searle that even allowing that the amount and type of computation are relevant is still to hold the position that computation alone is sufficient for thinking. Specifying the amount and type of computation is not to add anything to the computation. Besides, who has offered to draw a distinction between not enough computation and enough computation, or between the right kind and the wrong kind?

The second reply of Searle is that even with robot interaction with the world there will not be understanding because the input and output will still just be the manipulation of symbols. As has been remarked, the success of such a response of course assumes that Searle has already shown with the Chinese room argument that manipulation of symbols is not sufficient for understanding (Copeland, 1993, p. 132) [41]. His response here is thus parasitic on his earlier argument. As I have argued, even if Searle has not shown this, he may have called into question the analogy on which strong AI builds its argument that computation is sufficient for understanding, and so his response to the robot reply may succeed if this earlier argument has succeeded.

However, here we may be able to at last, sort out exactly what Searle thinks the computer is missing. A computer embodied in the form of a robot will be able to match some symbols (for example those composing names) with other symbols (the binary digits that it processes as it receives input from its television cameras, auditory devices, and so forth). Similarly, on the symbol or binary digit level, it will match certain instructions it gives with body movement input and other “sensory” input from its external environment. Now why would anyone advancing the robot reply think that this addition to the computer would result in understanding if its previous symbol manipulation? No matter how extensive and speedy did not? (Note that the robot reply tacitly acknowledges that Searle has been correct all along in thinking that the non-robot computer cannot understand! Searle does seem right about this.

By allowing that a computer must have some sort of robot body to understand, it concedes that syntax alone will not give the computer semantics. Searle should do more to play off one AI camp against the other. How can it be so obvious that computers do or will understand if the robot reply camp tacitly concedes that something more is needed?) The robot reply position must be that adding "sensory" input, etc. would provide for semantics instead of just syntax. Semantics has to do with the meaning of sentences, as opposed to the correctness of their structure grammatically. "Sensory" input would presumably allow the computer to match its words with things in the world, and thus attain the "word-world" connections needed in semantics. Therefore, to Searle's claim that the computer has syntax without semantics, the robot reply is that here is a way to get the semantics.

How can Searle still are so dissatisfied with the conceptual equipment of the computer-as-robot to continue to deny that it understands when it has just the word-world connections that would seem to provide what is, needed for semantics? Here is where we see that what Searle really may be driving at with his charge that computers lack intentionality and understanding is the claim that they lack any sort of conscious awareness. (Recall that intentionality refers to the fact that our beliefs and wishes, for example, are "about" something.) It seems that all any of us have in going from syntax to semantics is a word-world connection that such robots will have, except for one thing--conscious awareness of meaning. When we use the term "tree," we know what it means because we have the experience of trees in various ways, sensations of trees, of pictures of trees, etc. This comes through my sense organs or vicariously through someone else's sense organs. Now the robot can match its symbols for the term "tree" to other symbols for what comes through its "sense organs," in a way perhaps analogous to what goes on in humans. Why then would Searle not think this is enough for semantics? Because the robot has no conscious awareness of anything--trueness, tree sensations, etc. Any representation, any "intentionality" on the part of the robot is a purely "nonconscious" mapping of some symbols onto other symbols, and in understanding the meaning of words and sentences we do more than this. To Searle the robot has no original intentionality. This is why when he looks at the Chinese translation scenario involving a robot, with the entire word-world connections one could ask for, Searle still refuses to concede it understands. Symbol manipulation, whether mapping words to things or not, will not create conscious awareness.

It certainly seems as if at the root of Searle's problems with the notion of computer understanding is his belief that they have no conscious awareness. However, this is not entirely certain. First, he does not focus on conscious awareness as the missing extra when he talks of computers not being able to understand. If that is what it is, why not come out and say so? Second, in other places, he distinguishes between consciousness and intentionality, and the notion of understanding that computers do not have seems to be more a denial of intentionality to them. It is not obvious that all he means by intentionality is conscious representation. As well, since he sometimes equates the notion of a computer understanding with the notion of computer thinking, the claim that conscious awareness is a necessary component of understanding means that it is a necessary component of thinking, and if this is his view he cannot allow the possibility of unconscious thought. I'm not sure that he holds such a view.

What can we make of the plausibility of Searle's objection to the sufficiency of robot word-world connections for semantics and therefore understanding? Here we are back to the earlier discussion of Copeland's discussion of intentionality in the context of the Turing test.

We find some support for his view with Robinson. However, first we must note a distinction between word-world connections as a necessary condition of understanding and such connections as a sufficient condition of understanding. The robot reply argues that what is lacking in computers (and Searle's Chinese room) is word-world connections, and so providing the computer with these (via a robot body, etc.) is sufficient for understanding, given all else that is going on with the computer. Searle objects that since there would still be no understanding such word-world connections are not sufficient. Whether he is correct or not, this is a different issue from the one of whether such connections are necessary for understanding.

An argument that robots probably will not have conscious experience comes from Robinson, who asks us to imagine a series of robot scenarios that start with a simple computer. In each scenario the computer gains more sophistication, a body, more integration, etc. The point of all these robot scenarios is to try to flush out our intuitions about whether adding various body associations, including processing of external input and the ability to act, enable it to experience pain, understand, and have its actions mean something to it. A crucial point is reached with robot “Frp,” who is, embodied, and can act in response to commands, for example “Go to the drugstore and bring back the prescription that the druggist will have for me.” By carrying out such a command, does this embodied, acting robot show that it understands? (Recall that Searle would reply that such activity does not show that it understands, and furthermore it cannot understand because it is just manipulating symbols without the presence of semantic meaning). Robinson's answer is that it depends on what you mean by the term “understanding.” Here Robinson distinguishes between understanding in the sense connected to appropriate action and understanding in the sense connected to sensation and feelings. In the first sense of “understanding,” Robinson thinks it does understand, because it engages in appropriate action. This is a functional sense of understanding--the robot can engage in the appropriate output given a particular input, and so we can say that it understands. Not understanding in this sense would be if it failed to produce the desired output, as if you told it to go to the drugstore, and instead it went to the supermarket. (Clearly, this sense of understanding would not satisfy Searle.) However, in the second sense of “understanding,” robot Functional Reactive Programming (FRP), who has no sensations or feelings, does not understand. According to Robinson, this is because having no sensations and feelings, its actions ultimately have no point for it (Robinson, 1992, pp. 39-54) [37]. This is somewhat of an odd way to put it. We would have expected Robinson to say that it lacks this kind of understanding because it has no conscious awareness.

Note that: *Functional Reactive Programming* or FRP is a paradigm for programming hybrid systems (i.e., systems containing a combination of both continuous and discrete components) -- in a high-level, declarative way. The key ideas in FRP are its notions of continuous, time-varying values, and time-ordered sequences of discrete events. FRP is, used to express two programming language features critical to robotics: time flow, as characterized by both continuous- and discrete-time signals, and reaction, allowing a system to reconfigure itself in response to stimulus.

Functional Reactive Programming is a programming paradigm, which extends functional programming with values that change over time. FRP systems can be, classified into continuous and discrete FRP depending on the way time is, modeled.

So it seems that the only sense of understanding relevant to robots is the weak sense of performance, not the strong sense involving sensations and feelings. The strong sense of

understanding does seem to involve some sort of conscious awareness, for it is the sense in which a congenitally blind person, never having experienced the color of anything, does not understand “Chlorophyll is green” and “Strawberries are red” (Robinson, 1992, p. 53) [37]. This is what Chalmers would call a phenomenal sense of understanding, as opposed to a psychological sense. Robinson's positions then supports Searle's claim that even if a robot has input and output involving the external world, if it does not have sensations and feelings as we know them, then it will not understand in any strong phenomenal sense. Word-world connections will give it semantic meaning in the weak, functional sense of understanding, but there is no reason to believe, and reason to doubt, that it is enough to give it semantics in the sense of conscious awareness of meaning and therefore understanding in the strong sense.

We conclude that while Searle may not have proved that computers and robots have no understanding in the sense of conscious awareness, his story might give us pause in merely assuming they do. A robot provided with a sophisticated sensory capacity for interacting with the world may provide for semantics and not just syntax. However, whether this is enough for it to be consciously aware of its surroundings is an open question. The question of robots and consciousness will continue in the next chapter.

REFERENCES

[1] Paul, G., & Cox, E. (1996). *Beyond humanity: Cyberevolution and future minds*. Rockland: Charles River Media.

[2] Searle, J. (1980). Minds, brains, and programs, In D. Hofstadter & D. Dennett (Eds.), *The Mind's I* (pp. 353-373). New York: Basic Books.

[3] Kurzweil, R. (1999), The Age of spiritual machines. New York: Viking.

[4] Moravec, H. (1999), Robot: Mere machine to transcendent mind. New York: Oxford University Press.

[5] Moore, G. (1965), Cramming more components onto integrated circuits. *Electronics* 38, 8, 114-117.

[6] Schaller, R. (1996), The Origin, nature, and implications of “MOORE'S LAW” [Online]. Available: http://research.microsoft.com/~Gray/Moore_Law.html [1999, Aug 26].

[7] Schaller, R. (1997). Moore's law: Past, present, and future. *IEEE Spectrum*, 34, 6, 53-59.

[8] Schaller, R. (1996). *The Origin, nature, and implications of “MOORE'S LAW”* [Online]. Available: http://research.microsoft.com/~Gray/Moore_Law.html [1999, Aug 26].

[9] Moore, G. (1965) Cramming more components onto integrated circuits. *Electronics 38,* 8, 114-117.

[10] Rosenberg, S. (1999). *Will the real Moore's law please stand up?* [Online]. Available: http://www.salon1999.com/21st/rose/1997/10/02straight.html [1999, Aug 10].

[11] Kurzweil, R. (1999). *The Age of spiritual machines*. New York: Viking.

[12] Moore's law. (1995). [Online]. Available: http://www.netmeg.net/jargon/terms/m/moore_s_law.html [1999, November 2].

[13] Moore, G. (1997). An Update on Moore's law [Online]. Available: http://www.intel.com/pressroom/archive/speeches/gem93097.htm [1999, Aug 26].

[14] Kane, M. (1997). [Letter to the editor] *IEEE Spectrum,* 34, 8, 8.

[15] Meieran, E. (1998). 21st Century semiconductor manufacturing capabilities [Online]. Available: http://developer.intel.com/technology/itj/q41998/articles/art_1.htm [1999, Aug 28].

[16] Rosenberg, S. (1999). Will the real Moore's law please stand up? [Online]. Available: http://www.salon1999.com/21st/rose/1997/10/02straight.html [1999, Aug 10].

[17] Kanellos, M. (1999). Moore says Moore's law to hit wall [Online]. Available: http://www.news.com/News/Item/0,4,14751,00.html [1999, Aug 10].

[18] Kanellos, M. (1999). Moore says Moore's law to hit wall [Online]. Available: http://www.news.com/News/Item/0,4,14751,00.html [1999, Aug 10].

[19] Geppert, L. (1998), The Media event: Moore's law mania. IEEE Spectrum, 35, 1, 20-21.

[20] Lemos, R. (1999) Chips to hit size barrier in 2012 [Online]. Available: http://www.zdnet.co.uk/news/1999/25/ns-8637.html [1999, Aug 10].

[21] Rupley, S. (1997). Defying Moore's law [Online]. Available: http://www.zdnet.com/pcmag/news/trends/t970922a.htm [1999, Aug 26].

[22] Stix, G. (1996), Waiting for breakthroughs. *Scientific American, 274,* 4, 94-99.

[23] Feynman, R. (1959). There's plenty of room at the bottom [Online]. Available: http://nano.xerox.com/nanotech/feynman.html [1999, October 21].

[24] Copeland, J. (1993), Artificial intelligence: A philosophical introduction. Oxford: Blackwell.

[25] Moravec, H. (1988). *Mind children: The future of robot and human intelligence*. Cambridge: Harvard University Press.

[26] Yam, P. (1998). Intelligence considered. Scientific American Quarterly [Online]. Available: http://www.sciam.com/specialissues/1198intelligence/1198yam.html [1999, Oct 6].

[27] Sternberg, R., & Detterman, D. (Eds.). (1986). What is intelligence? Norwood: Ablex Publishing Corporation.

[28] Kelly, J. (1993). *Artificial intelligence: A modern myth*. New York: E. Horwood.

[29] Sternberg, R. (1990). *Metaphors of mind*. Cambridge: Cambridge University Press.

[30] Sternberg, R., & Detterman, D. (Eds.). (1986). *What is intelligence?* Norwood: Ablex Publishing Corporation.

[31] Gottfredson, L. (1998). The General intelligence factor. *Scientific American Presents: Winter 1998: Exploring Intelligence*, 24-29.

[32] Gardner, H. (1998). A Multiplicity of intelligences. *Scientific American Presents: Winter 1998: Exploring Intelligence*, 18-23.

[33] Wilson, L. (1998b), The eighth intelligence: Naturalistic intelligence [Online]. Available: http://www.uwsp.edu/acad/educ/lwilson/LEARNING/natintel.htm [1999, Aug 26].

[34] Haugeland, J. (1986), *Artificial intelligence: The very idea*. Cambridge: MIT Press.

[35] Turing, A. (1950). Computing machinery and intelligence. In D. Hofstadter & D. Dennett (Eds.), *The Mind's I* (pp. 53-67). New York: Basic Books.

[36] Gandy, R. (1996). Human versus mechanical intelligence. In P. Millican, & A. Clark (Eds.), *Machines and thought* (pp. 125-136). Oxford: Oxford University Press.

[37] Robinson, W. (1992). *Computers, minds & robots*. Philadelphia: Temple University Press.
[38] Copeland, J. (1993). *Artificial intelligence: A philosophical introduction*. Oxford: Blackwell.
[39] Searle, J. (1980). Minds, brains, and programs. In D. Hofstadter & D. Dennett (Eds.), *The Mind's I* (pp. 353-373). New York: Basic Books.
[40] Hofstadter, D. (1981). Reflections, In D. Hofstadter & D. Dennett (Eds.), *The Mind's I* (pp. 373-382). New York: Basic Books.
[41] Schaller, R. (1997). Moore's law: Past, present, and future. *IEEE Spectrum*, *34*, 6, 53-59.

APPENDIX A: FLUORESCENCE MICROSCOPY

A fluorescence microscope is an optical microscope that uses fluorescence and phosphorescence instead of, or in addition to, reflection and absorption to study properties of organic or inorganic substances.

A.1. INTRODUCTION

The "Fluorescence Microscope" refers to any microscope that uses fluorescence to generate an image, whether it is a more simple set up like an epifluorescence microscope, or a more complicated design such as a confocal microscope, which uses optical sectioning to get better resolution of the fluorescent image. See Figure A-1.

Fluorescent dyes are a critical component of the systems described below because they allow for specific and sensitive determination of the localization of molecules (i.e., proteins and DNA) within the cell. Simply, fluorescent dyes are themselves molecules that are able to absorb light of one wavelength and then emit light of another, longer wavelength. For fluorescence microscopy, these dyes are usually bound to the cellular molecule under study. By illuminating a dye-labeled specimen with light matching the excitation spectrum of the dye, and then collecting the emitted light, it is possible to visualize only the location of the dye molecules (providing specificity). Figure A-4 includes a schematic of an epifluorescence microscope like the one used to collect the image data used below and shows how the exciting light and emitted light are separated. Another important feature of fluorescent dyes is that they are visualizable in relatively low numbers (providing sensitivity). Although the systems described below are not so sensitive, specially constructed microscope systems have been able to visualize single dye molecules.

The specificity of fluorescent dyes is partially a secondary characteristic, however. That is, each dye must be targeted and bound to a particular cellular molecule in order to work as a specific marker. One way to accomplish this targeting is to conjugate the dye to an antibody. Antibodies are proteins produced by the immune systems of higher organisms and are capable of binding very specifically and strongly to their target molecule (the antigen), usually a protein. Humans, for instance, are producing between 10^7 and 10^9 *different* antibodies at any given time.

In general, antibodies to be used for immunofluorescence are obtained by first injecting a mouse with a sample of the protein to be labeled. After allowing time for the mouse to mount

an immune response to the injected, foreign protein, antibody-producing cells are collected from the mouse's spleen. Since these cells will not grow indefinitely in vitro, they are fused with tumor cells so that the resulting hybrid cells both produce antibodies and proliferate indefinitely (such a cell line is called a hybridoma). From among all of the cells that successfully fuse, individual cells are then cloned (i.e., allowed to proliferate in their own dish). Because a mouse will have immune cells producing a variety of antibodies at any given time, each of the clones is then screened to determine whether it produces antibodies against the desired protein. After all of these steps, one is left with a hybridoma that produces many copies of the same antibody molecule. The antibodies produced from such a hybridoma are called monoclonal because the cells producing the antibodies are all derived from a single clone. Figure A-2 depicts the major steps in this process. The Figure A-2 is reprinted from MOLECULAR CELL BIOLOGY by Lodish et al. ©1986, 1990, 1996 by Scientific American Books, Inc. Used with permission by W.H. Freeman and Company.

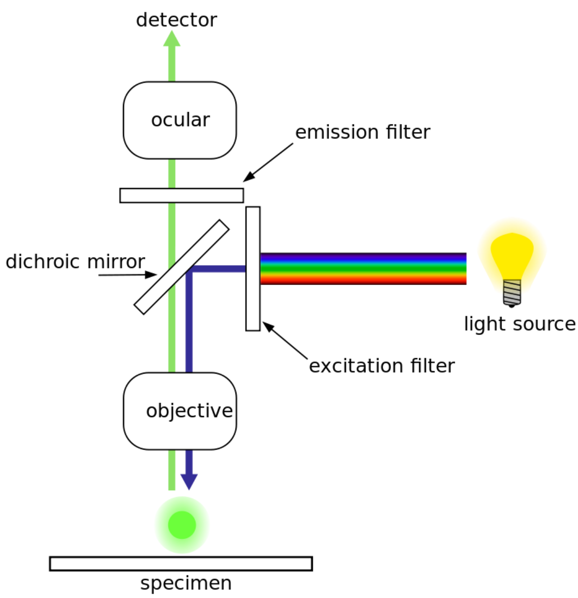


Figure A-1. Schematic of a Fluorescence Microscope.

Once monoclonal antibodies are available for the desired protein, there are two general approaches to attaching a fluorescent dye to each antibody. The first approach is to chemically conjugate the dye molecules directly to the antibodies (*direct* immunofluorescence-see Figure A-3). While this method allows the fluorescent labeling of a specimen to be accomplished in a single step, it also requires each kind of antibody to be conjugated with dye in separate steps. A second approach relies on the fact that all antibodies produced by a mouse have a common region in their protein structure (the F_c region). It is therefore possible using another kind of animal, say a goat, to generate antibodies against the F_c region of mouse antibodies. The fluorescent dye can then be conjugated to a large quantity of this secondary antibody. The now fluorescent secondary antibody is then used to label each of the primary antibodies which in turn label their antigenic protein. This method

of indirect immunofluorescence (see Figure A-3) was used for most of the antibody labeling in the work described below. Both direct and indirect immunofluorescence are able to utilize the high specificity of antibody-antigen binding and the sensitivity of fluorescence microscopy.

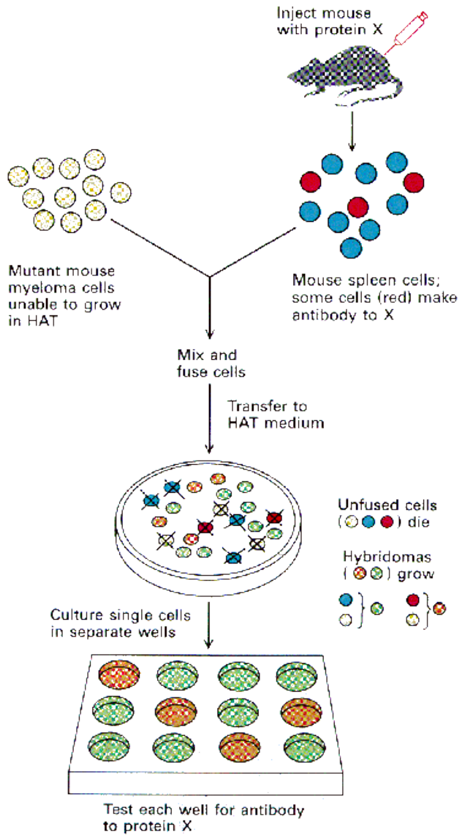


Figure A-2. Schematic Representation of the Process Used to Generate Monoclonal Antibodies Against a Protein of Interest.

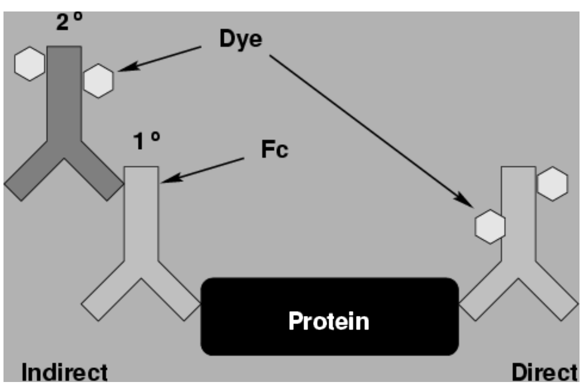


Figure A-3. Schematic Representation of Indirect (left) and Direct (right) Immunofluorescence.

After the cells have been labeled appropriately, they are taken to a fluorescence microscope (see Figure A-4) for imaging. One important aspect of a fluorescence microscope is the effective resolution of the image it is able to create (i.e., the smallest distance two fluorescent objects must be separated by in order for them to be resolved). Analysis of both the transverse (in the image plane) and axial resolution of the microscope are detailed below.

No matter how well designed, a microscope objective cannot collect all of the light emitted by a fluorescent specimen. Because of this, and because the wavelength of the fluorescence is finite, high frequency information is lost and the transverse resolution of a fluorescence microscope is finite. Theoretically, the transverse resolution is defined by the Rayleigh criterion which says that the minimum distance (D) between two point sources in the specimen, such that they can still be resolved, is equal to the radius (r) of the Airy disk for that microscope system. Simply, an Airy disk is the output of the microscope system corresponding to a point source at the input. The radius of the Airy pattern is defined by

$$r = \frac{1.22\lambda}{2NA} \qquad \text{Eq. A-1}$$

where λ is the wavelength of the emitted light in air and NA is the numerical aperture of the lens ($NA = n\sin\theta$ where n is the refractive index of the medium between the specimen and the lens, and θ is the half-cone angle of light captured by the objective) [1, p. 31]. Following this analysis for λ =520 nm, and NA=1.3, the approximate transverse resolution (D) of the microscope system used in this work is 200 nm.

Given that the data below were collected as stacks of images by changing the focus after each slice, the axial resolution of the system is also important. The axial resolution is determined by the depth of field of the microscope, and is defined by

$$\Delta_z = \frac{2\lambda n}{NA^2} \qquad \text{Eq. A-2}$$

where λ , n, and NA are as defined above. For λ =520 nm, NA=1.3, and n=1.5, $\Delta_z \approx$ 920 nm, or about 5 times worse than the transverse resolution. For comparison, the cells used below are $\approx$100-150 µm across.

A.2. EPIFLUORESCENCE MICROSCOPY

The majority of fluorescence microscopes, especially those used in the life sciences, are of the epifluorescence design shown in the diagram. Light of the excitation wavelength is focused on the specimen through the objective lens. The fluorescence emitted by the specimen is focused to the detector by the same objective that is used for the excitation which for greater resolution will need objective lens with higher numerical aperture. Since most of the excitation light is transmitted through the specimen, only reflected excitatory light reaches the objective together with the emitted light and the epifluorescence method therefore gives a

high signal-to-noise ratio. The dichroic beam splitter acts as a wavelength specific filter, transmitting fluoresced light through to the eyepiece or detector, but reflecting any remaining excitation light back towards the source.

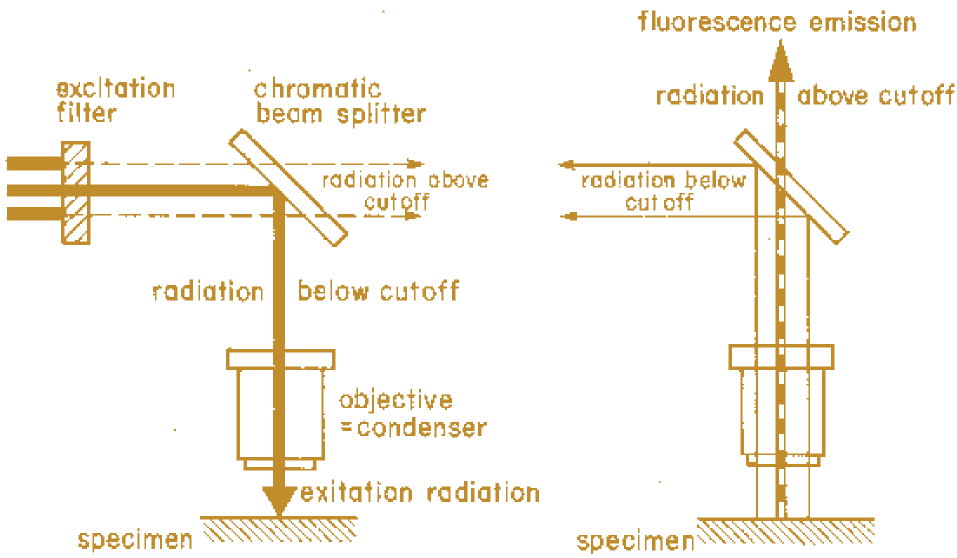


Figure A-4. A Schematic of the Light Path in an Epifluorescence Microscopy.

Figure A-4 is illustration of excitation light (left) is passed through the excitation filter and then reflected off the dichroic beam splitter. This light then passes through the objective to illuminate the specimen. Light emitted from the specimen (right) is collected by the objective and then passed through the dichroic and an emission filter (not shown) before being collected via an eyepiece or camera (also not shown). Reprinted, with permission of the publisher, from Fluorescence Microscopy of Living Cells in Culture: Part A. Fluorescent Analogs, Labeling Cells, and Basic Microscopy, Vol. 29 of Methods in Cell Biology, ©1989 by Academic Press, Inc.

A.3. Principle of Fluorescence Microscope

The specimen is illuminated with light of a specific wavelength (or wavelengths) which is absorbed by the fluorophores, causing them to emit light of longer wavelengths (i.e., of a different color than the absorbed light). The illumination light is separated from the much weaker emitted fluorescence through the use of a spectral emission filter. Typical components of a fluorescence microscope are a light source (xenon arc lamp or mercury-vapor lamp are common; more advanced forms are high-power LEDs and lasers), the excitation filter, the dichroic mirror (or dichroic beam splitter), and the emission filter (see figure below). The filters and the dichroic beam splitter are chosen to match the spectral excitation and emission characteristics of the fluorophore used to label the specimen [2]. In this manner, the distribution of a single fluorophore (color) is imaged at a time. Multi-color images of several types of fluorophores must be composed by combining several single-color images [3].

Most fluorescence microscopes in use are epifluorescence microscopes, where excitation of the fluorophore and detection of the fluorescence are done through the same light path (i.e., through the objective). These microscopes are widely used in biology and are the basis for more advanced microscope designs, such as the Confocal Microscope and the Total Internal Reflection Fluorescence Microscope (TIRF).

A.4. CONFOCAL MICROSCOPY

Confocal microscopy, most frequently Confocal Laser Scanning Microscopy (CLSM), is an optical imaging technique for increasing optical resolution and contrast of a micrograph by means of adding a spatial pinhole placed at the confocal plane of the lens to eliminate out-of-focus light [2]. It enables the reconstruction of three-dimensional structures from sets of images obtained at different depths (a process known as optical sectioning) within a thick object. This technique has gained popularity in the scientific and industrial communities and typical applications are in life sciences, semiconductor inspection and materials science.

A conventional microscope "sees" as far into the specimen as the light can penetrate, while a confocal microscope only "sees" images one depth level at a time. In effect, the CLSM achieves a controlled and highly limited depth of focus.

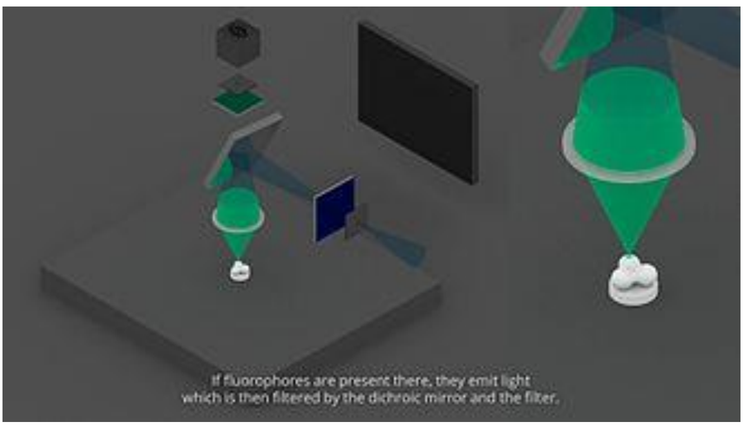


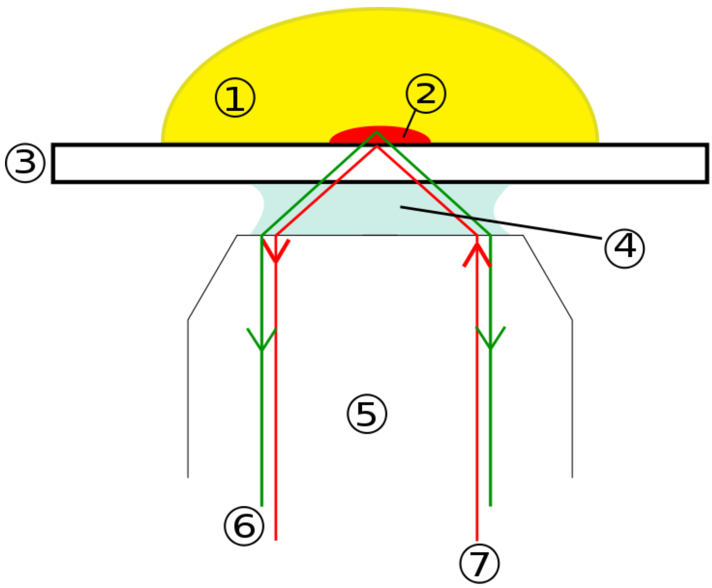
Figure A-5. Fluorescent and Confocal Microscopes Operating Principle.

Note that, depth of focus is a lens optics concept that measures the tolerance of placement of the image plane (the film plane in a camera) in relation to the lens. In a camera, depth of focus indicates the tolerance of the film's displacement within the camera, and is therefore sometimes referred to as "lens-to-film tolerance."

A.5. TOTAL INTERNAL REFLECTION FLUORESCENCE MICROSCOPY

A Total Internal Reflection Fluorescence Microscope (TIRFM) is a type of microscope with which a thin region of a specimen, usually less than 200 nm can be observed.

The idea of using total internal reflection to illuminate cells contacting the surface of glass was first described by E.J. Ambrose in 1956 [4]. This idea was then extended by Daniel Axelrod [5] at the University of Michigan, Ann Arbor in the early 1980s as TIRFM. A TIRFM uses an evanescent wave to selectively illuminate and excite fluorophores in a restricted region of the specimen immediately adjacent to the glass-water interface. The evanescent wave is generated only when the incident light is totally internally reflected at the glass-water interface. The evanescent electromagnetic field decays exponentially from the interface, and thus penetrates to a depth of only approximately 100 nm into the sample medium. Thus the TIRFM enables a selective visualization of surface regions such as the basal plasma membrane (which are about 7.5 nm thick) of cells as shown in the figure above. Note, however, that the region visualized is at least a few hundred nanometers wide, so the cytoplasmic zone immediately beneath the plasma membrane is necessarily visualized in addition to the plasma membrane during TIRF microscopy. The selective visualization of the plasma membrane renders the features and events on the plasma membrane in living cells with high axial resolution.



1. Specimen
2. Evanescent wave range
3. Cover slip
4. Immersion oil
5. Objective
6. Emission beam (signal)
7. Excitation beam

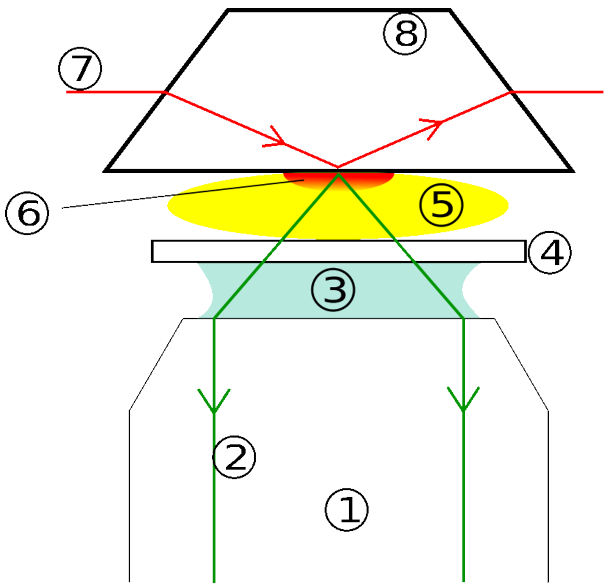
Figure A-6. (Cis-) Total Internal Reflection Fluorescence Microscopy (TIRFM) Diagram.

TIRF can also be used to observe the fluorescence of a single molecule, [5-6] making it an important tool of biophysics and quantitative biology.

Cis-geometry (through-objective TIRFM) and trans-geometry (prism- and light guide based TIRFM) have been shown to provide different quality of the effect of total internal reflection. In the case of trans-geometry, the excitation light path and the emission channel

are separated, while in the case of objective-type TIRFM they share the objective and other optical elements of the microscope. Prism-based geometry was shown to generate clean evanescent wave, which exponential decay is close to theoretically predicted function [6]. In the case of objective-based TIRFM, however, the evanescent wave is contaminated with intense stray light. The intensity of stray light was shown to amount 10-15% of the evanescent wave, which makes it difficult to interpret data obtained by objective-type TIRFM.

The (Trans) Total Internal Reflection Fluorescence Microscope (TIRFM) diagram.



1. Objective
2. Emission beam (signal)
3. Immersion oil
4. Cover slip
5. Specimen
6. Evanescent wave range
7. Excitation beam
8. Quartz prism

Figure A-7. Illustration of (Trans-) Total Internal Reflection Fluorescence Microscope (TIRFM) Diagram.

REFERENCES

[1] Shinya Inoué and Kenneth R. Spring, *Video Microscopy*: *The Fundamentals*, Plenum Press, New York and London, 2 edition, 1997.

[2] S. Nie, D. T. Chiu, and R. N. Zare, "Probing individual molecules with confocal fluorescence microscopy," *Science*, vol. 266, no. 5187, pp. 1018-21, 1994.

[3] Spring KR, Davidson MW. "Introduction to Fluorescence Microscopy." *Nikon Microscopy*. Retrieved 2008-09-28.

[4] Ambrose, E. J. (24 Nov 1956). “A surface contact microscope for the study of cell movements.” *Nature*. 178 (4543): 1194. Bibcode: 1956Natur.178.1194A. doi:10.1038/1781194a0.

[5] Axelrod, D. (1 April 1981). “Cell-substrate contacts illuminated by total internal reflection fluorescence.” *The Journal of Cell Biology*. 89 (1): 141–145. doi:10.1083/jcb.89.1.141. PMC 2111781Freely accessible. PMID 7014571.

[6] Yanagida, Toshio; Sako, Yasushi; Minoghchi, Shigeru (10 February 2000). “Single-molecule imaging of EGFR signaling on the surface of living cells.” *Nature Cell Biology*. 2 (3): 168–172. doi:10.1038/35004044. PMID 10707088.

[7] Ambrose, W; et al. (1999). “Single-molecule detection with total internal reflection excitation: comparing signal-to-background and total signals in different geometries.” *Cytometry*. 36(3): 224.

INDEX

A

B

C

D